Robust Linear Parameter Varying Control of an Unmanned Aerial Vehicle

Thesis submitted for the degree of

Doctor of Philosophy

at the University of Leicester

by

Jianchi Chen

Department of Engineering

University of Leicester

February 2010

Author

Jianchi Chen

Robust Linear Parameter Varying Control of an Unmanned Aerial Vehicle

Abstract

The dynamic response characteristics of modern unmanned aerial vehicles (UAVs) are highly nonlinear and vary substantially with flight conditions due to their reduced dimensions compared to normal aircraft. In this thesis, design frameworks that are based on parameter dependent Lyapunov functions (PDLFs) are developed for UAV flight control systems. These design frameworks or procedures can systematically deal with aircraft systems with nonlinear and parameter dependent dynamics, and uncertainty in the mathematical models. To this end, we analyse robust stability and performance of LPV systems and present two LPV controller design methods using the PDLF approach: Two-Degree-of-Freedom (2DoF) and loop shaping with coprime factorisation. We formulate and solve the control problem for an LPV plant with measurable parameters and an output feedback structure. The solvability conditions are reduced to LMIs and can be solved approximately using finite-dimensional convex programming. A parameter dependent performance approach is used in a 2DoF/PDLF design and constitutes a flexible generalisation for calibrations of local performance. In loop shaping/PDLF design, a left coprime factorisation is derived by H_2 filtering, and then a loop shaping design is implemented in the PDLF framework. We also incorporate pole placement constraints into the LMI synthesis to improve controller performance. To be able to use the robust gain-scheduling synthesis results, an LPV model of the UAV is developed and validated. The gain scheduling controller design of longitudinal/lateral-directional dynamics of the UAV is illustrated in the design example. It is shown that a flight control system can be built with satisfactory robust stability and performance.

Contents

List of Figures

List of Tables

Notation and Acronyms

Mathematical Symbols

$:=$	defined by
§	section
$\mathbf{R}$	set of real numbers
$\mathbf{R}^n$	the set of n-dimensional real vector
$\mathbf{R}^{n \times m}$	set of n by m matrices with element in $\mathbf{R}$
I_n	identity matrix of size n
$\mathcal{C}^1(U, V)$	set of continuously differentiable functions from U to V
$\mathrm{trace}(M)$	the trace of square matrix M
$\| \cdot \|_2$	induced L_2 norm
ρ	parameter vector
$\mathcal{P}$	compact parameter set
$\mathcal{F}_{\mathcal{P}}$	parameter variation set
γ	optimal performance index in terms of L_2 gain

Symbols for Aircraft Dynamics

I_x, I_y, I_z	moments of inertia about x, y and z body axes
I_{xy}	product of inertia with respect to x and y body axes
F_x, F_y, F_z	total forces along x, y and z body axes
X_{cg}	centre of gravity on reference chord
M_s	mass of the aircraft
C_M	pitch moment coefficient
C_L	lift coefficient
α	angle of attack, deg
h	altitude
u, v, w	axial, lateral and normal velocity, m/s
V_t	total velocity, m/s
β, γ_f	angle of sideslip and flight path angle, deg
p, r	roll rate and yaw rate, deg/s

ϕ, ψ	roll angle and heading angle
η, τ	elevator deflection, throttle input
ξ, ζ	aileron and rudder deflection
θ	pitch angle
P_N	north position in earth axis
P_E	east position in earth axis

Acronyms

LFT	Linear Fractional Transformation
LFR	Linear Fractional Representation
LMI	Linear Matrix Inequality
LTI	Linear Time Invariant
LPV	Linear Parameter Varying
MIMO	Multiple Input Multiple Output
2DOF	Two Degree of Freedom
SQLF	Single Parameter Dependent Lyapunov Function
PDLF	Parameter Dependent Lyapunov Function

Acknowledgements

I am very grateful to my supervisor Professor Ian Postlethwaite. His guidance during my doctoral work has been invaluable. I appreciate his support, patience and encouragements throughout these years. It has been an honour and a pleasure to work with him.

I would also like to thank my co-supervisor Professor Da-Wei Gu for his help and useful discussions during my study.

I would like to thank Dr Emmanuel Prempain for inspiring discussions. His insights and knowledge of LPV control definitely helped me out in many related topics. I would also like to thank Dr. Prathyush Menon for helpful discussions and useful tips on the simulations. And I want to thank Dr. Kannan Natesan for helping me out on many problems with his solid knowledge of aircraft dynamics.

Thanks to the other members of the Control and Instrumentation Research Group, including Dr. Michael Chen, Dr. Ping Li, Dr. Liqun Yao, Dr. Zhe Yang, who have made my stay here enjoyable.

The University of Leicester Department of Engineering has provided excellent research facilities, making it possible for me to conduct my research.

Finally, I am greatly indebted to my wife Le Fang, for her patience, understanding and dedication to our small home here in Leicester, and to our parents, who have supported me throughout my studies with their encouragement and love.

To my parents
my wife Le,
and my son Evan, with all my love.

Chapter 1

Introduction

1.1 Background

The design of reliable flight control systems for aircraft has been under research for decades. In recent years, the development of flight controllers for unmanned aerial vehicles (UAVs) has started to receive attention with their increasing civil and military applications. The methodologies that have been previously applied to robust flight control systems for UAVs are H_∞ control [1], adaptive control [2], μ synthesis [61] and many others. UAVs offer advantage over conventional manned vehicles in many ways because they can be used in environments otherwise too dangerous for normal vehicles. Furthermore, without the need for a pilot, UAVs are also capable of handling long surveillance missions. The development of high performance UAVs will see an increasing need to perform high offensive flight missions whilst the dynamics of the vehicle are not well known. This is mainly because the reduced dimensions of these vehicles lead to highly nonlinear system behaviours and unconventional dynamics in terms of natural frequency and damping. Furthermore, the sensitivity to changes in flight conditions (concerning velocity more than altitude), the assumptions in aerodynamic models(e.g. stability and control derivatives) and the inaccuracies in geometric and inertial data represent a set of uncertainties in plant and environment modellings. The result is that the conventional flight control design methods are often not very effective. Therefore, the design of a robust and high performance flight control system which guarantees a suitable level of robustness to environmental changes and modelling inaccuracies plays a key role in the development of UAVs.

During the past decade, there has been an increased interest in developing the gain-scheduling control technique for nonlinear and time varying systems (e.g. aircraft) which greatly stimulates the development of *Linear Parameter Varying* (LPV) control. Gain-scheduling is a method that designs LTI controllers for systems with widely varying nonlinear or parameter dependent dynamics at different operating points. The linear controller gains are then adjusted (scheduled) in real time as the operating conditions vary. There are many successful applications of this control strategy, including jet engines [27], missile autopilots [28] and V/STOL aircraft [29].

The classic approach to this problem consists of repeated designs associated with some scheduling strategy connecting locally designed controllers. Such a scheme, however lacks supporting theories that guarantee the robust behaviour of the scheduled controllers. Gain-scheduled designs are mostly guided by heuristic rules. The most fundamental guidelines are

1. the scheduling variables should vary slowly, and
2. the scheduling variables should capture the plant's nonlinearities.

These guidelines remind us that designs were based on LTI approximations of the actual plant. Thus, these approximations must be sufficiently accurate if one is to expect the local feedback properties to be carried over to the overall scheduled system. However, the approaches used frequently in gain-scheduling design might not be sufficient for multivariable controller design. In [30], some potential hazards involved in traditional gain scheduling are pointed out. Much attention has since turned to analysing and designing parameter dependent controllers for LPV systems. In [9], Sharhruz and Behtash suggested using LPV systems for gain-scheduling controller synthesis. In [21], an LPV integrated flight controller is designed for a flexible aircraft. Robustness analysis of a gain-scheduled LPV flight control system using integral quadratic constraints is studied in [22]. A significant contribution to the theory of LPV control is the formulation of gain-scheduling problems in the context of convex optimisation, expressed in the form of Linear Matrix Inequalities (LMIs). LMIs have now become a very natural mechanism for the formulation of H_∞ gain-scheduling problems based on pioneering work of Gahinet and Apkarian [5, 16].

One way to apply LPV control is to obtain Linear Fractional Transformations (LFTs) [51] of the nonlinear model and use small-gain theorem and scaled H_∞ optimisation to provide closed-loop stability and induced L_2 performance [7, 8]. Another path that leads to gain-scheduling controllers is the search for adequate Lyapunov functions that establish stability and performance bounds for the closed-loop system. Becker [31] applied the notion of quadratic stability to LPV systems, and he extended it to quadratic performance by adding sufficient conditions for bounding the induced L_2 norm. These conditions are characterised by a single quadratic Lyapunov function (SQLF) that satisfies certain LMIs, and quadratic performance is provided using linear parameter-dependent feedback. The LFT gain-scheduling [16] and quadratic gain scheduling [17] techniques make use of SQLF to characterise stability and performance.

This SQLF method has been found to be conservative because the parameters are expected to vary arbitrarily quickly, even though in reality scheduling parameters cannot vary arbitrarily fast. Wu and others [18, 23, 46] have reduced the conservatism of this LMI-based control, though for a greater computational cost, by using parameter dependent Lyapunov functions (PDLFs). Wu's method allows a priori bound on the parameter's rate of variation, and then solves the control synthesis problems by reformulating existence conditions into a semi-infinite dimensional convex optimisation. The goal of the control is to stabilise the parameter dependent closed-loop system and to provide disturbance/error attenuation as measured by the induced L_2 norm. In [20], Bennani employed PDLFs to explicitly take bounds on the rate of change of the parameters into account and synthesised a parameter dependent controller for a missile benchmark problem by scaled differential LMIs. PDLF has also been used by Gahinet [19] and Feron [47] to study the robustness of LTI systems to constant real, parametric uncertainties.

1.2 Motivation

In this thesis, two control system design procedures, 2DoF/PDLF and loop shaping/PDLF, will be presented. They are particularly suitable for designing flight controllers, as they can systematically deal with multi-input multi-output (MIMO) plants with nonlinear and parameter-dependent dynamics. Designers can also build performance specifications into the plant by adjusting weighting functions for controller synthesis. The motivation for developing these methods will be given in this section.

There are many controller design applications that use parameter dependent Lyapunov functions [15, 31], where a constant performance index is applied in the synthesis. In [34], it was pointed out that a uniform performance index should be considered as worst-case (or "global") with respect to both perturbations and parameter trajectories. It is suggested that the performance index should be generalised to an integral quadratic constraints (IQC). The performance function should then be optimised at each operating point of the plant. In this way, a varying performance index can be obtained without sacrificing superior performance at relatively benigh operating points for some "troublesome" operating points. In chapter 4, we discuss some issues involving the implementation of the IQC method, and propose a parameterisation method which is more efficient and easier to implement. We incorporate parameter-dependent performance index into the 2DoF/PDLF design framework and show in the design example of chapter 6 that we are able to achieve better robust performance when compared to the methods with a constant performance index.

Building on the renowned work of McFarlane and Glover [37], the loop shaping design methodology has gone through various successful applications [13, 38, 53]. It is a popular H_∞ design procedure that has the advantage of offering levels of robust performance based on proper selection of weighting function matrices. The choice of the weighting matrices depends on various factors, such as bandwidth, roll-off rates, and low frequency gain magnitude. Some recent developments of loop shaping design using LMI approaches can be found in [40, 44], where a single quadratic Lyapunov function (SQLF) approach is adopted using either a normalised coprime factorisation or a four-block framework to design static loop shaping controllers. Loop shaping design for LPV systems can be found in [41], where Glover and Jung design a calibratable loop shaping controller for an LPV model of a diesel engine. Papageorgiou [39] has also applied the LPV loop shaping method to aircraft systems. Prempain [43] introduced a novel method, where the determination of a left coprime factorisation of a polytopic system can be formulated as a special H_2 filtering problem. In [42], a gain-scheduling loop shaping design procedure for a benchmark LPV missile model is presented. A polytopic model of the plant is converted into a left coprime factorisation structure using the method in [43], and then static and full order loop shaping controllers are designed by solving a system of LMIs. In chapter 4, we will extend the loop shaping design method for LPV systems in [42] using a PDLF approach, and specifically Wu's frameworks in [15, 23]. We propose a design and synthesis procedure and it will be shown in chapter 6 that we can design a robust gain-scheduling controller using the loop shaping/PDLF method with better robust performance than previous designs using the SQLF method.

Like the classic H_∞ methodology, PDLF synthesis has very little control over transient behaviour and closed-loop pole locations. It would therefore be desirable if it were possible to incorporate pole placement constraints into the PDLF design to provide control over transient behaviour. A useful tool is the notion of the LMI region introduced in [25, 24], where pole assignment in convex regions of the left-half plane can also be expressed as LMI constraints on Lyapunov matrices. In chapter 4, we will fit the concept of the LMI region into Wu's PDLF synthesis frameworks. The tests for pole placement constraints are numerically tractable as they are in LMI form. By assigning the closed-loop poles in pre-specified regions, a well-damped, fast-decaying and practical controller dynamic can be imposed on a PDLF controller synthesis for a UAV. This will be demonstrated in chapter 7.

1.3 Organisation

The remainder of this thesis can be organised as follows:

Chapter 2 reviews some important results that will be used in the later chapters, including linear matrix inequality, coprime factorisation uncertainty and loop shaping design procedure (LSDP).

Chapter 3 analyses the stability and parameter dependent performance issues of LPV systems in terms of parameter dependent Lyapunov functions. A rigorous definition for the class of varying parameters and LPV systems is given. We discuss the notion of quadratic stability and define an induced L_2 norm performance objective for LPV systems. We define the γ performance output feedback problem and show how to obtain a simplified LPV system for the PDLF synthesis. Finally, the LMI solvability conditions for SQLF and PDLF control synthesis are given and subsequently we will discuss some computational issues of the LMI synthesis.

Chapter 4 presents two design methods using parameter dependent Lyapunov functions, 2DoF/PDLF and loop shaping/PDLF. In the 2DoF/PDLF design, we synthesise a controller for the LPV plant with parameter dependent γ performance. We analyse the problems associated with the previous method and propose a simple parameterisation to account for the parameter dependent performance. The controller design procedure is also given. For the loop shaping/PDLF design method, we use a robust H_2 filtering method to obtain a left coprime factorisation of the LPV plant. We reformulate Wu's assumptions and prove that the loop shaping problem can be fitted nicely into Wu's PDLF framework. We prove the solvability conditions for the loop shaping/PDLF method and describe the design procedure. In the last section, we incorporate pole placement constraints into the output-feedback PDLF synthesis by adding LMI constraints into the original LMI solvability conditions for the PDLF synthesis.

Chapter 5 considers UAV modelling. A general form of coupled nonlinear flight dynamic equations of motion is first presented. Then the longitudinal and lateral-directional dynamics of the UAV are given. We transform the nonlinear model into an LPV model using Jacobian linearisations around a group of equilibrium points. We further validate the LPV model by time response simulations. Finally, we discuss the dynamics and parameter dependence of the UAV model.

Chapter 6 demonstrates the flight controller design methods by their applications to the “Eclipse” UAV demonstrator. Parameter dependent longitudinal and lateral controllers are designed using both the 2DoF/PDLF and loop shaping/PDLF methods. Some practical considerations of robust performance, pole locations and computations will be investigated. The tracking responses of these two controllers are demonstrated in the nonlinear simulations.

Chapter 7 provides a summary of contributions and some concluding remarks. Directions for future work are also given.

In the Appendix, the geometric data of the “Eclipse” UAV are given.

1.4 Contributions

The main contributions of the thesis can be summarised as follows:

- We discuss robust stability and performance of LPV systems in a PDLF framework. We review the solvability conditions of output feedback synthesis and discuss various issues regarding the implementations of the PDLF synthesis.

- We develop two control system design procedures for LPV systems, 2DoF/PDLF and loop shaping/PDLF. In a PDLF framework, parameter dependent Lyapunov functions can exploit the bounded parameter variation information and reduce the conservatism caused by single quadratic Lyapunov functions. This point is justified in the design example.

- A parameter dependent performance index is adopted in the 2DoF/PDLF design. We analyse drawbacks of the IQC method in the previous literature and instead of implementing a weighting function approach to adjust the IQC performance level using an iterative procedure, we propose a direct parameterisation method which is more flexible and efficient. We treat the parameter dependent performance as a function of scheduling parameter. Compared to the iterative IQC method, we only need to compute the optimal performance once and the number of the Lyapunov variables is also greatly reduced. In this way, we are able to achieve better robust performance in the design example when compared to the PDLF design that uses a uniform bound on the performance index.

- We formulate the H_∞ loop shaping design procedure in a PDLF framework. The assumptions of the PDLF synthesis are modified to allow the use of a left coprime factorisation configuration which is derived by a robust H_2 filtering technique. The solvability conditions for the loop shaping/PDLF synthesis are presented with proof. In the design example, we investigate the effects of model uncertainties on the performance of the loop shaping/ PDLF controller.

- The concept of convex $\mathcal{D}$-stable regions is incorporated into the PDLF synthesis. We define a vertical strip convex region and prove that the poles of the closed-loop system can be placed in such a region by adding some LMI constraints. In the design example, we show that the undesirable closed-loop pole locations can be eliminated by imposing the pole placement constraints.

- We develop a reliable LPV model from the nonlinear model of the UAV by Jacobian linearisation. The LPV model is then validated by simulations.

- The dynamics and stability of the UAV are analysed. We also investigate the parameter dependence of the nonlinear model and use the results in the design example. In this way, the rule of choosing the basis functions for Lyapunov variables (i.e. imitating the parameter dependence) is then justified by comparing the results of different parameterisations of the Lyapunov variables.

- We apply the two robust gain-scheduling controller design procedures to a flight control system design for the “Eclipse” UAV. The performance and stability of the gain-scheduled controller are evaluated with full flight envelope conditions. The simulation results show satisfactory tracking response and robust performance.

Chapter 2

Background and Preliminary

This chapter is comprised of some basic definitions and elementary results. While the treatment of these materials is by no means exhaustive, it is sufficient to use as a reference for the theoretical developments in the later chapters.

2.1 Linear Matrix Inequality

In recent years, Linear Matrix Inequalities (LMIs) has emerged as an powerful tools for solving a number of control problems. LMIs approach is able to formulate an optimisation problem in a form of linear matrix inequality constraints [3]:

$$F(x) := F_0 + \sum_{i=1}^{n} x_i F_i > 0, \tag{2.1}$$

where $F_i = F_i^T \in \mathbf{R}^{n \times n}$ are explicit given. The minimisation problem can be stated as:

$$\text{minimize} \quad c^T x \text{subject to} \quad F(x) > 0.$$

The numerical solution for the above convex minisation problem can be efficiently derived by using interior point method [6] while an analytical or closed-form solution might be difficult to find. The LMIs approaches are well suited to formulate many problems that we encounter in control engineering especially in the domain of robust control which usually deals with the system robustness with the present of modelling and measurement errors. Based on the description these uncertainties in the system and data, we are able to construct a convex optimization problem that yields a robust solution. Furthermore, the formulation of the LMI constraints enable us to analyse the trade-off of the various performance specification and add multi-criterion constraints in the controller synthesis procedure, which offer great advantage over the other control methods that can only rely on single criterion to reflect all the design constraints. The LMIs approach has proved to be very useful in robust control design and achieve various successful applications [17, 18, 36, 44]. Here we will present some important results that will be used in the later chapters.

Lemma 1 (**Schur Complement**, [3]) Partition the $(m+n)\times(m+n)$ matrix M appropriately as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}.$$

If M_{22} is invertible, then

$$\det(M) = \det(M_{22})\det(M_{11} - M_{12}M_{22}^{-1}M_{21}).$$

Similarly, if M_{11} is invertible, then

$$\det(M) = \det(M_{11})\det(M_{22} - M_{21}M_{11}^{-1}M_{12}).$$

If M is symmetric (i.e., M_{11} and M_{22} are symmetric and $M_{21} = M_{12}^T$), then the following statements are equivalent:

1. M is positive-definite.
2. $M_{22} > 0$ and $M_{11} - M_{12}M_{22}^{-1}M_{12}^T > 0$
3. $M_{11} > 0$ and $M_{22} - M_{12}^T M_{22}^{-1} M_{12} > 0$

Lemma 2 (**Matrix Inversion Lemma**, [4]) For appropriately dimensioned matrices A, B, C and D

$$(A \pm BCD)^{-1} = A^{-1} \mp A^{-1}B(C^{-1} \pm DA^{-1}B)^{-1}DA^{-1},$$

assuming that A and C are invertible.

Lemma 3 (**Gahinet & Apkarian**, [5]) Given a symmetric matrix $R = R^T \in \mathbf{R}^{n\times n}$ and matrices $U, V \in \mathbf{R}^{m\times n}$, let $U_\perp$ and $V_\perp$ be the matrices whose columns form bases for the kernels of U and V, respectively. There exists a matrix $K \in \mathbf{R}^{m\times m}$ satisfying

$$R + U^T K V + V^T K^T U < 0$$

if and only if

$$U_\perp^T R U_\perp < 0 \text{ and } V_\perp^T R V_\perp < 0$$

Proof: See [5].

Lemma 4 (**Bounded Real Lemma**, [5]) Consider an nth-order LTI system $G(s)$ with state-space realization

$$\begin{bmatrix} \dot{x}(t) \\ d(t) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix}.$$

$G(s)$ is asymptotically stable and $\|G(s)\|_\infty < \gamma, \gamma > 0$ if and only if there exists $W \in \mathbf{R}^{n\times n}$ such that

$$\begin{bmatrix} A^T W + WA & WB & C^T \\ B^T W & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0.$$

Lemma 5 (Packard, [7]) Given a pair of positive-define matrices $X, Y \in \mathbf{R}_+^{n\times n}$, there exist $X_2, Y_2 \in \mathbf{R}^{n\times r}$ such that

$$\begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix} > 0 \text{ and } \begin{bmatrix} Y & Y_2 \\ Y_2^T & Y_3 \end{bmatrix} = \begin{bmatrix} X & X_2 \\ X_2^T & X_3 \end{bmatrix}^{-1}$$

if and only if

$$\begin{bmatrix} Y & I_n \\ I_n & X \end{bmatrix} \geq 0 \text{ and rank}\left(\begin{bmatrix} Y & I_n \\ I_n & X \end{bmatrix}\right) \leq n + r$$

Proof: See [7]; this lemma can be proved by using the Schur complement and matrix inversion lemma.

2.2 Loop Shaping Design Techniques

In this section, we will review the Loop Shaping Design Procedure (LSDP). The objective of loop shaping design is to produce a controller that guarantees robust stability against normalised coprime factor uncertainty. The coprime factorisation is used in LSDP to model the general stable perturbations in the plant [10, 37]. The development of the mathematics of robust stabilisation using H_∞ theory is somewhat involved, and a full mathematical description is not essential to understand the design process. Thus, only the main results will be given here. The original work of Glover and McFarlane can be found in [10, 37]. More details about the applications can be found in [11, 45]

2.2.1 Coprime Factor Uncertainty Representation

To illustrate the concept behind comprime factor uncertainty, consider a linear model transfer function $G(s)$ which can be factored as

$$G(s) = \tilde{M}(s)^{-1}\tilde{N}(s),$$

where $\tilde{M}(s)$ and $\tilde{N}(s)$ are stable, normalised left coprime transfer functions, i.e. there exist stable U, V such that

$$N(s)U(s) + M(s)V(s) = I.$$

Similarly a right coprime factorisation is given as $G(s) = N(s)M^{-1}(s)$, and there exist stable $U^*(s)$ and $V^*(s)$ that satisfy $U^*(s)N(s) + V^*(s)M(s) = I$. The right and left coprime factor realisations are

$$\begin{bmatrix} M \\ N \end{bmatrix} = \left[\begin{array}{c|c} A+BF & BS^{-\frac{1}{2}} \\ \hline F & S^{-\frac{1}{2}} \\ C+DF & DS^{-\frac{1}{2}} \end{array}\right] \tag{2.2}$$

$$[\tilde{M} \quad \tilde{N}] = \left[\begin{array}{c|cc} A+LC & L & B+LD \\ \hline R^{-\frac{1}{2}}C & R^{-\frac{1}{2}} & R^{-\frac{1}{2}}D \end{array}\right] \tag{2.3}$$

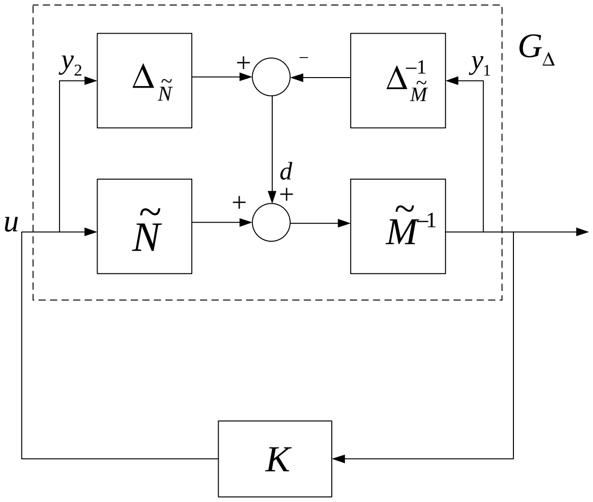


Figure 2.1: Normalised left coprime factor plant

where

$$
\begin{aligned}
F &= -S^{-1}(D^TC + B^TX)\\
L &= -(BD^T + ZC^T)R^{-1}\\
R &= I + DD^T\\
S &= I + D^TD
\end{aligned}
$$

This representation is particularly useful because the coprime factor representation contains no unstable hidden mode. Therefore it can be used to represent an unstable transfer function by two stable factors. A perturbed plant G_Δ as shown in Figure 2.1 can be represented by

$$G_\Delta := (\tilde M + \Delta_{\tilde M})^{-1}(\tilde N + \Delta_{\tilde N}),$$

where $\Delta_{\tilde M}$ and $\Delta_{\tilde N}$ are left coprime factor perturbations or uncertainties, and the H_∞ norm of the perturbations is bounded by

$$\|[\Delta_{\tilde M} \ \ \Delta_{\tilde N}]\| < \epsilon.$$

where $\epsilon > 0$ is the stability margin. A stabilisation problem of the nominal plant G is to find a controller K which stabilises the perturbed plant G_Δ and maximises the stability margin.

For the perturbed feedback system in Figure 2.1, it can be shown that

$$\begin{bmatrix} y_1 \\ -y_2 \end{bmatrix} = \begin{bmatrix} (I+GK)^{-1}\tilde{M}^{-1} \\ K(I+GK)^{-1}\tilde{M}^{-1} \end{bmatrix} d$$

and that

$$d = -\begin{bmatrix} \Delta_{\tilde{M}} & \Delta_{\tilde{N}} \end{bmatrix} \begin{bmatrix} y_1 \\ -y_2 \end{bmatrix}.$$

As the norms of the perturbations are bounded by ϵ, then a robust stabilisation problem is to find a stabilising controller that satisfies

$$\|T_{yd}\|_\infty = \left\| \begin{matrix} (I+GK)^{-1}\tilde{M}^{-1} \\ K(I+GK)^{-1}\tilde{M}^{-1} \end{matrix} \right\|_\infty = \left\| \begin{matrix} S_o\tilde{M}^{-1} \\ KS_o\tilde{M}^{-1} \end{matrix} \right\|_\infty < \epsilon^{-1} := \gamma$$

where $S_o = (I+GK)^{-1}$ is the output sensitivity function. This realisation can be seen as a special case of the H_∞ mixed-sensitivity design. The standard problem of mixed-sensitivity design can be solved iteratively. But Glover and MacFarlane have shown that, if the above normalised coprime uncertainty is used, the optimal value of ϵ and γ can be found directly without iteration as

$$\gamma_{min} = \epsilon_{max} = \left\{1 - \|\tilde{N} \ \ \tilde{M}\|_H^2\right\}^{-\frac{1}{2}} = (1+\rho(XZ))^{\frac{1}{2}},$$

where $\|\cdot\|_H$ denotes the Hankel norm and ρ denotes the spectral radius (maximum eigenvalues). Z and X are the solutions to the generalised control algebraic Riccati equations

$$(A - BS^{-1}D^TC)Z + Z(A - BS^{-1}D^TC)^T - ZC^TR^{-1}CZ + BS^{-1}B^T = 0$$

and

$$(A - BS^{-1}D^TC)^TX + X(A - BS^{-1}D^TC) - XBS^{-1}B^TX + C^TR^{-1}C = 0$$

where

$$S = I + D^TD, \quad R = I + DD^T.$$

Thus, the robust stabilisation problem of finding K and ϵ_{max} reduces to the solution of two Riccati equations. The controller can be obtained directly as

$$K = \left[\begin{array}{c|c} A + BF + \gamma^2L^{-T}ZC^T(C+DF) & \gamma^2L^{-T}ZC^T \\ \hline B^TX & -D^T \end{array}\right] \tag{2.4}$$

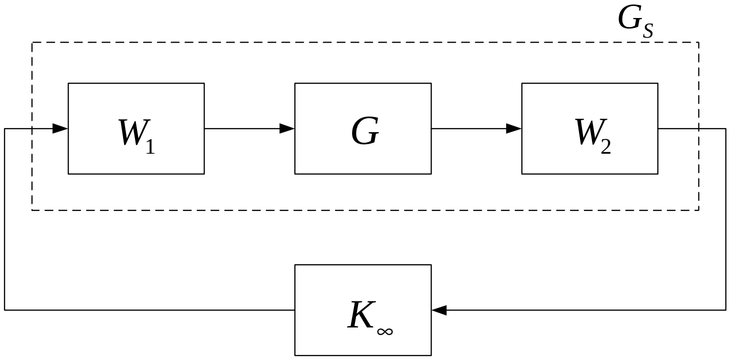


Figure 2.2: G is shaped by W_1 and W_2 and stabilised by K_∞

where

$$L = I + XZ - \gamma^2 I$$
$$F = -S^{-1}(D^T C + B^T X)$$

and (A, B, C, D) are the minimum realisation of G_Δ.

2.2.2 Loop Shaping Design Procedure

The robust stabilisation itself does not give much room for specifying the performance requirements. To do this McFarlane and Glover [37] divided the design process into two stages. First, the open-loop plant is augmented by weighting matrices to give desired shapes to the singular values of the open-loop frequency response. Then, the resulting shaped plant is robustly stabilised with respect to coprime factor uncertainty.

An implementation structure for the resulting H_∞ loop shaping controller can be derived from Figure 2.2. The nominal model of open-loop plant nominal model is augmented by pre- and post-compensators. The weighting matrix $W_1(s)$ is chosen to add integral action to boost the low frequency gain for good command tracking and ensure reasonable roll-off rates for the open-loop singular values around the desired crossover frequencies. The nominal model G and shaping functions $W_1(s)$ and $W_2(s)$

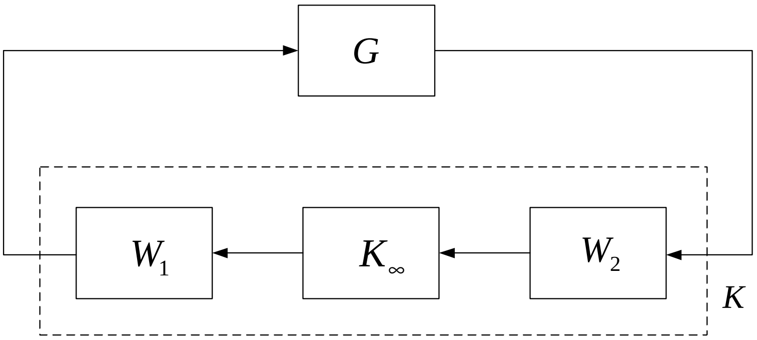


Figure 2.3: Final controller K is constructed by combining K_∞ with W_1 and W_2

are then combined to form the shaped model $G_s(s)$

$$G_s(s) = W_2(s)G(s)W_1(s)$$

The shaped system G_s is then robustly stabilised with respect to the left coprime factor uncertainty using H_∞ optimisation. An H_∞ controller $K_\infty(s)$ can be synthesised using the controller formula in (2.4). The final feedback controller K is then constructed by combining $K_\infty(s)$ with the shaping functions $W_1(s)$ and $W_2(s)$ as shown in Figure 2.3, such that

$$K = W_1(s)K_\infty(s)W_2(s)$$

Loop shaping design has a number of advantages, particularly for aerospace applications, which can be summarised below:

- It is relatively easy to implement, and the solution of the H_∞ loop shaping controller synthesis can be found without recourse to γ iteration as in normal H_∞ control problems.

- LSDP provides robustness to unstable perturbations and uncertainty in the location of lightly damped resonant poles, by maximising robustness to nomalised coprime-factor uncertainty [12].

- The resulting controller can be written as an exact plant observer plus state feedback. This structure allows gain scheduling and handling of input limitations; more details of this implementation can be found in [13].

- No pole-zero cancellation happens between the plant and controller. Pole-zero cancellations are common in many H_∞ control problems.

Chapter 3

Analysis of LPV systems Using Parameter Dependent Lyapunov Functions

In this chapter, we will discuss some important results of stability and performance of LPV systems as described in [15, 31, 32, 33, 34, 35, 36]. First we will introduce a class of LPV systems and a parameter set. Then we will define a form of quadratic stability that is characterised by PDLF and satisfies certain LMIs. Next, we introduce an induced L_2 norm performance measure for LPV systems with bounded parameter variation rates. We will show how to simplify an LPV system for the γ performance output feedback synthesis and give solvability conditions in the form of LMIs.

3.1 Linear Parametrically Varying Systems

In this section, we will define a useful class of LPV systems. Recall that the motivation for modelling a physical system as an LPV system is that we have some real-time information about the parameters whose value determines the plant dynamics. Therefore, we will begin with defining the class of parameter trajectories that governs an LPV system.

Given a compact subset $\mathcal{P} \subset \mathbf{R}^s$, we define a class of piecewise-continuous, vector-valued time signals ρ whose range is given by $\mathcal{P}$. This is the same class of the parameter trajectories considered in [16, 17], in which single quadratic Lyapunov functions (SQLFs) are used and the parameters are allowed to vary arbitrarily fast.

To differentiate the treatments of robust stability and performance for PDLF and SQLF, *priori* bounds on the rates of variation can be defined as

$$|\dot{\rho}_i(t)| \leq \nu_i \; (i = 1, \ldots, n_\nu), \quad \forall t \geq 0 \tag{3.1}$$

which confines the parameter derivative to a hypercube in $\mathbf{R}^s$. We can generalise this compact set to be a convex polytope $\mathcal{V}$, such that ν_i denotes the vertices of the polytope. These generalisations enable us to include the rates of variation in the robust control synthesis and remove some conservatism that is imposed on the control design [15, 20]. We will give a complete definition of the parameter set and use the notation for the remainder of the chapter.

Defintion 1 *Consider a compact set $\mathcal{P} \subset \mathbf{R}^s$ and a convex polytope $\mathcal{V}$ with vertices $\nu_i (i = 1, \ldots, n_\nu)$, then the parameter set $\mathcal{F}_\mathcal{P}$ denotes the allowable trajectories*

$$\mathcal{F}_\mathcal{P} := \{\rho \in (\mathbf{R}_+, \mathbf{R}^s) : \rho(t) \in \mathcal{P}, \dot{\rho}(t) \in \mathcal{V}(\rho(t)) \;\; \forall t \geq 0\}$$

which consist of all piecewise trajectories ρ that have compact range $\mathcal{P}$ and derivatives confined in the polytope $\mathcal{V}$.

With this definition of parameter trajectories, let the class of associated LPV systems be denoted by G_ρ and governed by

$$\begin{bmatrix} \dot{x}(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t), \dot{\rho}(t)) & B(\rho(t), \dot{\rho}(t)) \\ C(\rho(t), \dot{\rho}(t)) & D(\rho(t), \dot{\rho}(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ d(t) \end{bmatrix} \tag{3.2}$$

with the following characteristics:

- $A : \mathbf{R}^s \rightarrow \mathbf{R}^{n \times n}$ is continuous.
- $B : \mathbf{R}^s \rightarrow \mathbf{R}^{n \times n_d}$ is continuous.
- $C : \mathbf{R}^s \rightarrow \mathbf{R}^{n_e \times n}$ is continuous.

- $D : \mathbf{R}^s \to \mathbf{R}^{n_e \times n_d}$ is continuous.

Since the matrix functions A, B, C and D are continuous, they must be norm bounded on the compact set $\mathcal{P}$. We define the state transition matrix of G_ρ as $\Phi_\rho(t, t_0)$. More precisely, given the system

$$\dot{x}(t) = A(\rho(t), \dot{\rho}(t))x(t),$$

the state transition matrix $\Phi(t, t_0)$ is the unique solution to

$$\dot{\Phi}_\rho(t, t_0) = A(\rho(t), \dot{\rho}(t))\Phi_\rho(t, t_0), \tag{3.3}$$

for the initial condition $\Phi_\rho(t, t_0) = I_n$.

3.2 Parameter Dependent Stability and Performance of LPV systems

In this section, we will define parameter dependent quadratic stability for LPV systems using quadratic, parameter dependent Lyapunov functions. It can be shown that parameter dependent quadratic stability implies a strong form of robust stability for LPV systems with bounded parameter variation rates and can be defined algebraically in terms of solutions to LMIs.

Defintion 2 *Given a compact set $\mathcal{P}$ and polytope $\mathcal{V}$ such as in Definition 1, and a function $A : \mathbf{R}^s \to \mathbf{R}^{n \times n}$, the function A is parameter-dependent and stable over $\mathcal{P}$ if there exists a continuous and differentiable function $P : \mathbf{R}^s \to \mathbf{R}_+^{s \times s}$, where $P > 0$ satisfying*

$$A^T(\rho, \dot{\rho})P(\rho) + P(\rho)A(\rho, \dot{\rho}) + \sum_{i=1}^{s}(\dot{\rho}_i \frac{\partial P(\rho)}{\partial \rho_i}) < 0 \tag{3.4}$$

for all $\rho \in \mathcal{P}$ and $|\dot{\rho}| \leq \nu_i$. If A is the system matrix in (3.2), then we say the LPV system in (3.2) is parameter-dependent and stable over $(\rho, \dot{\rho})$.

If the parameter variation rates are assumed to be unbounded (e.g. $\nu \to \infty(i = 1, \ldots, n_\nu))$, and P is restricted to be constant, then the notion of quadratic parameter dependence is reduced to quadratic stability in [14], and is given as

$$A^T(\rho)P + PA(\rho) < 0.$$

Lemma 6 *[15] Given a compact set $\mathcal{P}$ and an LPV system, we have*

$$\dot{x}(t) = A(\rho(t), \dot{\rho}(t))x(t),$$

where $\rho \in \mathcal{F}_{\mathcal{P}}$. If the function A is parametrically stable over $\mathcal{P}$, then there exist some $\alpha > 0$, such that

$$A^T(\rho, \dot{\rho})P(\rho) + P(\rho)A(\rho, \dot{\rho}) + \frac{dP(\rho)}{dt} < -\alpha I_n \tag{3.5}$$

for all trajectories $\rho \in \mathcal{F}_{\mathcal{P}}$.

This lemma shows that the left-hand side of (3.4) is uniformly negative definite by compactness and continuity. The following lemma shows that parameter dependent quadratic stability gives a strong form of robust stability.

Lemma 7 *[15] Given $\mathcal{P}$ and the LPV system in Eq.(3.2), if A is parameter dependent quadratically stable over $\mathcal{P}$, and λ_1 and λ_2 are the minimum and maximum eigenvalues of $P(\rho)$ over $\mathcal{P}$ respectively, then for any $\rho \in \mathcal{F}_{\mathcal{P}}$, the state transition matrix $\Phi_\rho(t, t_0)$ satisfies*

$$\|\Phi_\rho(t, t_0)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1}} \mathrm{exp}\left[-\frac{\alpha}{2\lambda_2}(\mathrm{t} - \mathrm{t_0})\right] \tag{3.6}$$

where $\Phi_\rho(t, t_0)$ is given in (3.3).

Proof 1 *[15] If A is parameter dependent quadratically stable over $\rho \in \mathcal{F}_{\mathcal{P}}$, there exist matrix functions $P(\rho)$ that satisfy 3.4. And by compactness, we have*

$$\lambda_1 I_n \leq P(\rho) \leq \lambda_2 I_n. \tag{3.7}$$

Define the parameter dependent Lyapunov function $V : R^n \times R^s \to R$ as

$$V(x, \rho) = x^T P(\rho)x.$$

Differentiate both sides of the function along the parameter trajectories to give

$$\frac{d}{dt}V(x, \rho) = x^T(t)\left[A(\rho, \dot{\rho})P(\rho) + P(\rho)A(\rho, \dot{\rho}) + \frac{dP}{dt}\right]x(t).$$

From (3.5) and the above equation, we have the inequality

$$\frac{d}{dt}V(x,\rho) \leq -\alpha\|x\|^2.$$

Using $V(x,\rho) = x^T P(\rho)x$ *and (3.7), we get*

$$\lambda_1\|x\|^2 \leq V(x,\rho) \leq \lambda_2\|x\|^2, \tag{3.8}$$

and $\dfrac{V(x,\rho)}{\lambda_2} \leq \|x\|^2$, *so*

$$\frac{d}{dt}V(x,\rho) \leq -\alpha\frac{V(x,\rho)}{\lambda_2}.$$

Integrating both sides of the above inequality, leads to

$$V(x(t)) \leq V(x(t_0)) + \int_{t_0}^{t} -\alpha\frac{V(s)}{\lambda_2}ds$$

Using Gronwall's inequality, with $V(x(t_0))$ *constant, we establish that*

$$V(x(t)) \leq V(x(t_0))\mathrm{exp}\left(\int_{\mathrm{t_0}}^{\mathrm{t}} -\frac{\alpha}{\lambda_2}\mathrm{ds}\right) = \mathrm{V(x(t_0))exp}\left[-\frac{\alpha}{\lambda_2}(\mathrm{t}-\mathrm{t_0})\right].$$

Using (3.8) and the above inequality, we get

$$\lambda_1\|x(t)\|^2 \leq \lambda_2\|x(t_0)\|^2\mathrm{exp}\left[-\frac{\alpha}{\lambda_2}(\mathrm{t}-\mathrm{t_0})\right].$$

Recall that $x(t) = \Phi_\rho(t,t_0)x(t_0)$ *for all* $x(t_0) \in \mathbf{R}^n$, *and we finally have*

$$\|\Phi_\rho(t,t_0)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1}}\mathrm{exp}\left[-\frac{\alpha}{2\lambda_2}(\mathrm{t}-\mathrm{t_0})\right]$$

Clearly the above lemma shows that the state transition matrix decays exponentially with time, which suggests that $\lim_{t\to\infty}x(t) = 0$. This result justifies a strong form of parameter dependent quadratic stability.

Next we will define parameter dependent quadratic performance for the LPV system in (3.2) in terms of an induced L_2 performance from the disturbances to the error signals. Parameter dependent quadratic performance can be generalised in the same way as parameter dependent quadratic stability. In [15], Wu defines the induced L_2 norm of a quadratically stable LPV system, and states a sufficient condition to guarantee that G_ρ is quadratically stable and achieves a prescribed induced L_2 norm performance objective. This condition is written in terms of LMIs whose solutions can be formulated as a convex feasibility problem.

Lemma 8 *[15] Given a parametrically-dependent stable LPV system G_ρ in (3.2), there exists a constant scalar $M > 0$, such that for zero initial conditions $x(0) = 0$,*

$$\|G_\rho\|_2 = \sup_{\rho \in \mathcal{F}_\mathcal{P}} \sup_{\|d\|_2 \neq 0, d \in L_2} \frac{\|e\|_2}{\|d\|_2} \leq M < \infty$$

for all $\rho \in \mathcal{F}_\mathcal{P}$. And $\lim_{t \to \infty} x(t) = 0$ for any $x(0) \in \mathbf{R}^n, \rho \in \mathcal{F}_\mathcal{P}$, where $\| \cdot \|_2$ denotes the L_2 norm.

The proof of this lemma can be found in [15]. It establishes the important input-output and asymptotic properties of parameter dependent quadratically stable LPV systems. The L_2 norm level for an LPV system represents the largest ratio of disturbance norm to error over the set of linear operators described by the LPV system.

In order to characterise the input-output performance measured by the induced L_2 norm for the LPV system, a well known result given in chapter 2 is the Bounded Real Lemma which is useful in developing sufficient conditions for robust performance of LTI systems. The Bounded Real Lemma provides a tight bound on the H_∞ norm of a given LTI system. By formulating the lemma using parameter dependent Lyapunov functions, we are able to give a sufficient condition for checking the parameter dependent induced L_2 norm of the LPV system.

Lemma 9 *[34, 35] Given the LPV system G_ρ in (3.2), positive $\gamma(\rho) \in \mathcal{C}^1(R, R)$ and assuming that the function A is parametrically stable over $\mathcal{P}$ for all $\rho \in \mathcal{F}_\mathcal{P}$, the system G_ρ is said to have parameter dependent quadratic performance of level $\gamma(\rho)$ if there are some matrix functions $W : R^s \to R^{n \times n}$ satisfying $W^T = W > 0$ and*

$$\begin{bmatrix} A^T(\rho)W(\rho) + W(\rho)A(\rho) + \sum_{i=1}^{s} \dot{\rho}_i \frac{\partial W(\rho)}{\partial \rho_i} & W(\rho)B(\rho) & C^T(\rho) \\ B^T(\rho)W(\rho) & -\gamma(\rho)I & D^T(\rho) \\ C(\rho) & D(\rho) & -\gamma(\rho)I \end{bmatrix} < 0. \qquad (3.9)$$

Lemma 10 *[34] Suppose the LPV system G_ρ has parameter dependent performance of level $\gamma(\rho)$, then G_ρ is stable and, given any parameter trajectory $\rho \in \mathcal{F}_\mathcal{P}$, we have*

$$\int_0^\infty \frac{1}{\gamma(\rho(t))} e^T(t)e(t)dt < \int_0^\infty \gamma(\rho(t))d^T(t)d(t)dt$$

for all nonzero $d \in L_2$, assuming zero initial condition.

This lemma implies that the L_2 norm bound of disturbances to error signals is less than $\gamma(\rho)$, where $\gamma(\rho)$ is a continuous and parameter dependent function. The bound provided by this lemma is conceptually less conservative than the quadratic performance, whose W is restricted to be constant and the bound holds for arbitrarily fast parameter variation rates.

Note that in the above lemma, the bound is not necessarily tight when compared to the previous treatments because γ is parameter dependent. This modification is inspired by the fact that the system should be able to achieve different levels of performance at different operating points. With the parameter dependent γ, we are at a more flexible position to consider the notion of local LPV γ performance, which offers obvious advantage over a gain scheduling controller. For instance, we can reduce the conservatism of the performance level in an operational region where the LPV system is less susceptible to the perturbations, and trade-offs in performance may be negotiated between different subsets of parameter space. This point will be discussed and demonstrated further in the later chapters.

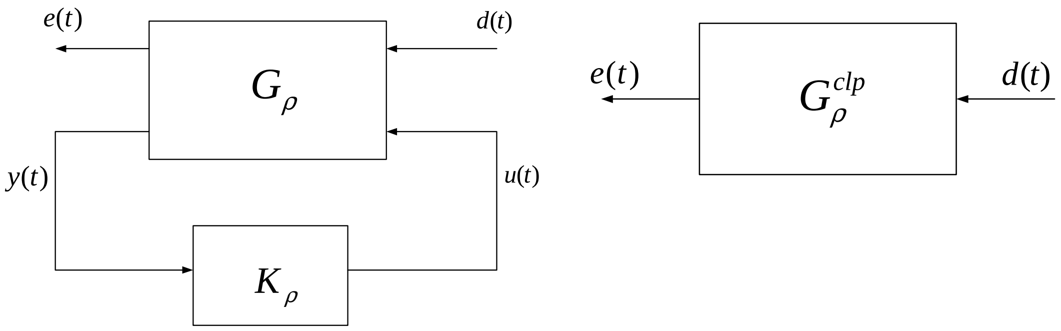
3.3 Simplified Open-loop LPV Systems for Induced L_2 Norm Control

In this section, we define a parameter dependent output-feedback problem for LPV systems. We show how to transform a standard LPV from into a simplified form for controller synthesis as described in [32] and [54].

Given a compact set $\mathcal{P} \in \mathbf{R}^s$, consider the open-loop LPV system G_ρ,

$$\begin{bmatrix} \dot{x}(t) \\ e(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B_1(\rho(t)) & B_2(\rho(t)) \\ C_1(\rho(t)) & D_{11}(\rho(t)) & D_{12}(\rho(t)) \\ C_2(\rho(t)) & D_{21}(\rho(t)) & D_{22}(\rho(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ d(t) \\ u(t) \end{bmatrix} \tag{3.10}$$

where $\rho(t) \in \mathcal{F}_{\mathcal{P}}$, and $x(t), \dot{x}(t) \in \mathbf{R}^n, d(t) \in \mathbf{R}^{n_d}, e(t) \in \mathbf{R}^{n_e}, u(t) \in \mathbf{R}^{n_u}, y(t) \in \mathbf{R}^{n_y}$. The matrix valued functions are of appropriate dimensions. The vector valued input/output signals in equation (3.10) have the following meanings: x and $\dot{x}$ represent



(a) a

(b) b

Figure 3.1: Closed-loop interconnection of an LPV system

the state of the system and its derivative, respectively; d represents exogenous inputs to the system, such as disturbances, measurements noise, and command references; e represents outputs defining control objectives, such as tracking error or command inputs; u is the control signal; y is the measured output of the system; and $\rho \in \mathcal{F}_{\mathcal{P}}$ are parameter trajectories of the LPV system .

We can define the m-dimensional parameterically dependent controller K_ρ as follows

$$\begin{bmatrix} u(t) \\ \dot{x}_K(t) \end{bmatrix} = \begin{bmatrix} D_K(\rho(t), \dot{\rho}(t)) & C_K(\rho(t), \dot{\rho}(t)) \\ B_K(\rho(t), \dot{\rho}(t)) & A_K(\rho(t), \dot{\rho}(t)) \end{bmatrix} \begin{bmatrix} y(t) \\ x_K \end{bmatrix} \tag{3.11}$$

with the continuous functions $A_K : \mathbf{R}^S \to \mathbf{R}^{m\times m}, B_K : \mathbf{R}^S \to \mathbf{R}^{m\times n_y}, C_K : \mathbf{R}^s \to \mathbf{R}^{n_u\times m}$ and $D_K : \mathbf{R}^s \to \mathbf{R}^{n_u\times n_y}$. The feedback controller can be applied to the open-loop system as shown in Figure 3.1, where the resulting closed-loop system is denoted G_ρ^{clp}. The controller synthesis problem can be described as follows: Given $\gamma > 0$, determine if there exists a ρ-dependent controller K_ρ, such that the closed-loop system G_ρ^{clp} is parameter dependent quadratically stable, and achieves an induced L_2 norm from disturbance input to error output less than γ for all $\rho \in \mathcal{F}_{\mathcal{P}}$.

To simplify the controller synthesis, we have made the following assumptions for the generalised plant (3.10) as in [15]:

(A1) $D_{22}(\rho) = 0_{n_y} \times 0_{n_u}$.

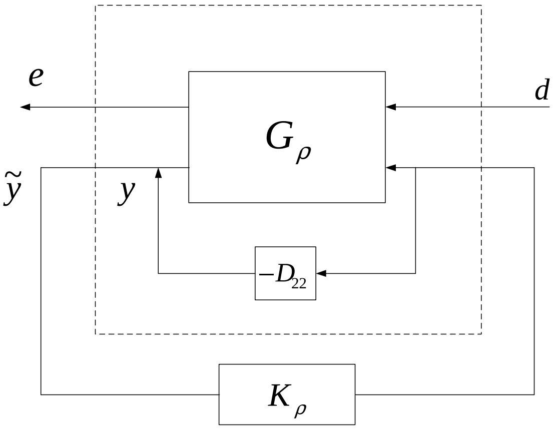


Figure 3.2: Eliminate the feed-through term in the synthesis

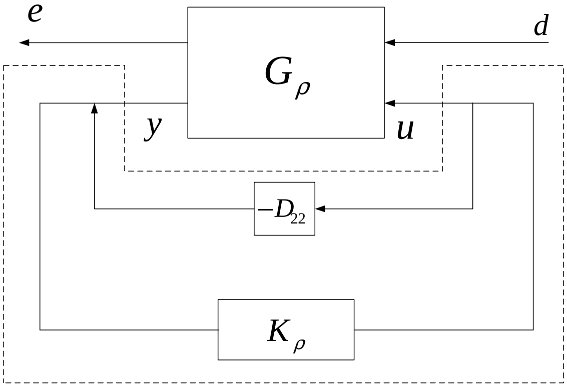


Figure 3.3: Include the feed-through term in the controller

(A2) $D_{12}(\rho)$ is full column rank for all $\rho \in \mathcal{P}$, and we can convert D_{12}^T to $[0 \;\; I_{n_y}]^T$.

(A3) $D_{21}(\rho)$ is full row rank for all $\rho \in \mathcal{P}$, and we can convert D_{21} to $[0 \;\; I_{n_u}]$.

Discussions about how to relax these assumptions can be found in [32]. Assumption A1 can be relaxed by including a feed-through term in the synthesis. To be more specific, for a plant that has $D_{22} \neq 0$, the output is given as

$$y(t) = C_2(\rho(t))x(t) + D_{21}(\rho(t))d(t) + D_{22}(\rho(t))u(t).$$

Define $\tilde{y}(t) := y(t) - D_{22}(\rho(t))u(t)$, then we can modify the original plant as shown in Figure 3.2, where the modified plant in the dashed box has a feed-through term $-D_{22}$. When the controller is applied to the original plant, the feed-through term should be added into the closed loop system as shown in the dashed box of Figure 3.3.

To satisfy assumptions A2 and A3, we can use some norm preserving transformations discussed in [54]. Under assumption A2, the rank of $D_{12}(\rho)$ is n_u for all $\rho \in \mathcal{P}$, then we know that the square root $[D_{12}^T(\rho)D_{12}(\rho)]^{\frac{1}{2}}$ is well defined. We denote $e_2 := n_u$ and $e_1 := n_e - e_2$. The matrix functions $Q_1 : \mathbf{R}^s \to \mathbf{R}^{n_e \times e_2}$ and $R_{12}(\rho) : \mathbf{R}^s \to \mathbf{R}^{e_2 \times e_2}$ can be defined as

$$\begin{aligned} Q_1(\rho) &= D_{12}(\rho)[D_{12}^T(\rho)D_{12}(\rho)]^{-\frac{1}{2}} \\ R_{12}(\rho) &= [D_{12}^T(\rho)D_{12}(\rho)]^{\frac{1}{2}}. \end{aligned}$$

It can be seen that for all $\rho \in \mathcal{P}$, $R_{12}(\rho)$ is invertible and $Q_1^T(\rho)Q_1(\rho) = I_{e_2}$. It is possible to find a matrix $Q_2(\rho) : \mathbf{R}^s \to \mathbf{R}^{n_e \times e_1}$ such that $Q_2^T(\rho)Q_1(\rho) := 0$ for all $\rho \in \mathcal{P}$, and $[Q_1(\rho) \;\; Q_2(\rho)]$ forms an orthonormal basis for $\mathbf{R}^{n_e \times n_e}$ for all $\rho \in \mathcal{P}$. The QR decomposition is a way to find such a Q_2. We define

$$Q_{12}(\rho) := \begin{bmatrix} Q_2^T(\rho) \\ Q_1^T(\rho) \end{bmatrix}.$$

Similarly, under assumption (A3) the rank of $D_{21}(\rho)$ is n_y for all $\rho \in \mathcal{P}$. Denote $d_2 := n_y$ and let $d_1 := n_d - n_{d_2}$. Let the matrix functions $\tilde{Q}_1 : \mathbf{R}^s \to \mathbf{R}^{n_d \times d_2}$, and $\mathbf{R}_{21}(\rho) : \mathbf{R}^s \to \mathbf{R}^{d_2 \times d_2}$ be defined as

$$\begin{aligned} \tilde{Q}_1(\rho) &= D_{21}^T(\rho)[D_{21}(\rho)D_{21}^T(\rho)]^{-\frac{1}{2}} \\ R_{21}(\rho) &= [D_{21}^T(\rho)D_{21}^T(\rho)]^{\frac{1}{2}}. \end{aligned}$$

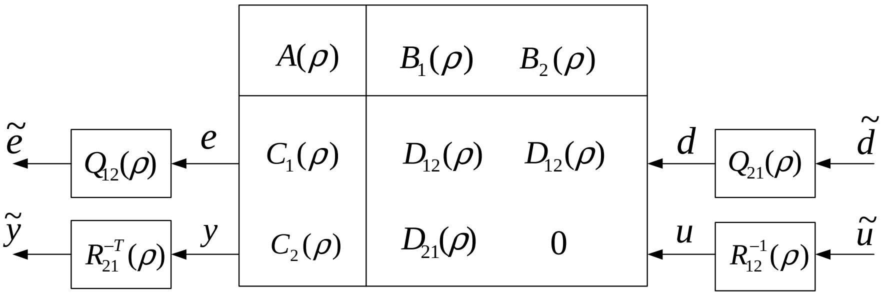


Figure 3.4: Scaled plant to obtain the simplified LPV model

For all $\rho \in \mathcal{P}$, R_{21} is invertible and $\tilde{Q}_1^T(\rho)\tilde{Q}_1(\rho) = I_{d_2}$. Define $\tilde{Q}_2(\rho)$ to be the continuous null space of $\tilde{Q}_1^T(\rho)$ and

$$Q_{21}(\rho) := \left[\tilde{Q}_2(\rho) \;\; \tilde{Q}_1(\rho)\right].$$

Using the above definitions for $Q_{12}, Q_{21}, R_{12}, R_{21}$, we can redefine the inputs and outputs of G_ρ as shown in Figure 3.4. The resulting LPV system can be partitioned by the following definitions:

$$[B_{11}(\rho) \;\; B_{12}(\rho)] := B_1(\rho)Q_{21}(\rho) = [B_1(\rho)\tilde{Q}_2(\rho) \;\; B_1(\rho)\tilde{Q}_1(\rho)],$$

$$\begin{bmatrix} C_{11}(\rho) \\ C_{12}(\rho) \end{bmatrix} := Q_{12}(\rho)C_1(\rho) = \begin{bmatrix} Q_2^T(\rho)C_1(\rho) \\ Q_1^T(\rho)C_1(\rho) \end{bmatrix},$$

$$\begin{aligned} \begin{bmatrix} D_{1111}(\rho) & D_{1112}(\rho) \\ D_{1121}(\rho) & D_{1122}(\rho) \end{bmatrix} &:= Q_{12}(\rho)D_{11}(\rho)Q_{21}(\rho) \\ &= \begin{bmatrix} Q_2^T(\rho)D_{11}(\rho)\tilde{Q}_2^T(\rho) & Q_2^T(\rho)D_{11}(\rho)\tilde{Q}_1^T(\rho) \\ Q_1^T(\rho)D_{11}(\rho)\tilde{Q}_2^T(\rho) & Q_1^T(\rho)D_{11}(\rho)\tilde{Q}_1^T(\rho) \end{bmatrix}, \end{aligned}$$

$$\tilde{B}_2(\rho) := B(\rho)R^{-1}(\rho), \qquad \tilde{C}_2(\rho) = R_{21}^{-T}(\rho)C_2(\rho).$$

$\tilde{D}_{12}(\rho)$ and $\tilde{D}_{21}(\rho)$ are given as

$$\tilde{D}_{12}(\rho) := Q_{12}(\rho)D_{12}(\rho)R_{12}^{-1}(\rho) \quad \tilde{D}_{12}(\rho) := R_{21}^{-T}(\rho)D_{12}(\rho)Q_{21}(\rho)$$

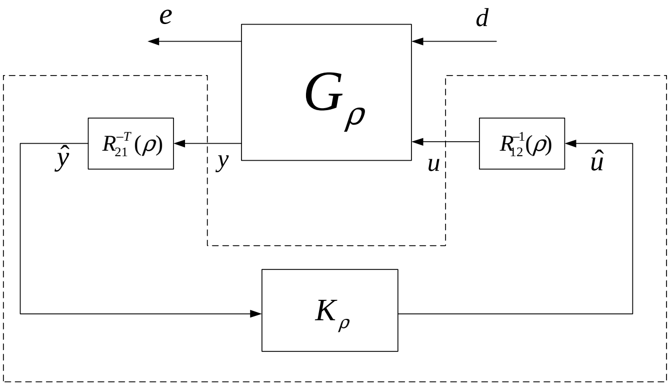


Figure 3.5: Scaled plant to obtain the simplified LPV model

From the definition of Q_{12}, R_{12}, we know that

$$\tilde{D}_{12}(\rho) = \begin{bmatrix} Q_2^T(\rho) \\ Q_1^T(\rho) \end{bmatrix} D_{12}[D_{12}^T \ \ D_{12}]^{-\frac{1}{2}} = \begin{bmatrix} Q_2^T(\rho) \\ Q_1^T(\rho) \end{bmatrix} Q_1(\rho) = \begin{bmatrix} 0 \\ I_{e_2} \end{bmatrix}.$$

Similarly, we have

$$\tilde{D}_{21}(\rho) = \left[D_{21}(\rho) \ \ D_{21}^T(\rho)\right]^{-\frac{T}{2}} D_{21}(\rho) \left[\tilde{Q}_2(\rho) \ \ \tilde{Q}_1(\rho)\right] = \tilde{Q}_1^T(\rho) \left[\tilde{Q}_2(\rho) \ \ \tilde{Q}_1(\rho)\right] = [0 \ \ I_{d_2}] \,.$$

With this simplified form of LPV system, we are then able to design a controller for the modified system. Since we use the norm preserving transformation on the disturbance and error signals, it is necessary to perform an inverse transformation when applying the controller to the system. We can apply R_{21}^{-T} and R_{12}^{-1} to the inputs and the outputs of the controller as depicted in Figure 3.5.

Assuming this type of transformation is possible for the open-loop LPV systems in the rest of the chapter, we are now able to define the simplified LPV system with performance objective. Given a compact set $P \in \mathbf{R}^s$, consider the open-loop LPV system G_ρ given as

$$\begin{bmatrix} \dot{x}(t) \\ e(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B_1(\rho(t)) & B_2(\rho(t)) \\ C_1(\rho(t)) & D_{11}(\rho(t)) & D_{12} \\ C_2(\rho(t)) & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ d(t) \\ u(t) \end{bmatrix}, \qquad (3.12)$$

where $\rho \in \mathcal{F}_{\mathcal{P}}$, and $x, \dot{x} \in \mathbf{R}^n, d \in \mathbf{R}^{n_d}, e \in \mathbf{R}^{n_e}, u \in \mathbf{R}^{n_u}$, and $y \in R^{n_y}$. We can further partition the state space data as

$$\begin{bmatrix} \dot{x}(t) \\ e_1(t) \\ e_2(t) \\ y(t) \end{bmatrix} = \left[\begin{array}{c|ccc} A(\rho(t)) & B_{11}(\rho(t)) & B_{12}(\rho(t)) & B_2(\rho) \\ \hline C_{11}(\rho(t)) & D_{1111}(\rho(t)) & D_{1112}(\rho(t)) & 0 \\ C_{12}(\rho(t)) & D_{1121}(\rho(t)) & D_{1122}(\rho(t)) & I_{e_2} \\ C_2(\rho(t)) & 0 & I_{d_2} & 0 \end{array}\right] \begin{bmatrix} x(t) \\ d_1(t) \\ d_2(t) \\ u(t) \end{bmatrix} \tag{3.13}$$

where $e_1 \in \mathbf{R}^{n_{e_1}}, e_2 \in \mathbf{R}^{n_{e_2}}, d_1 \in \mathbf{R}^{n_{d_1}}$ and $d_2 \in \mathbf{R}^{n_{d_2}}$, and the various state space matrices are dimensioned properly. Note that $n_{e_2} = n_u, n_{d_2} = n_y, n_{e_1} + n_{e_2} = n_e$ and $n_{d_1} + n_{d_2} = n_d$.

We assume that the parameter $\rho \in \mathcal{F}_{\mathcal{P}}$ is measurable in real time. Considering a finite-dimensional, linear, parameter and parameter rate dependent controller defined in (3.11), and defining $x_c := \begin{bmatrix} x^T & x_K^T \end{bmatrix} \in \mathbf{R}^{n+n_k}$, we can express the closed-loop system as

$$\begin{bmatrix} \dot{x}_{clp}(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} A_{clp}c(\rho(t), \dot{\rho}(t)) & B_{clp}(\rho(t), \dot{\rho}(t)) \\ C_{clp}(\rho(t), \dot{\rho}(t))) & D_{clp}(\rho(t), \dot{\rho}(t)) \end{bmatrix} \begin{bmatrix} x_{clp}(t) \\ d(t) \end{bmatrix} \tag{3.14}$$

where

$$\begin{aligned} A_{clp}(\rho, \dot{\rho}) &= \begin{bmatrix} A(\rho) + B_2(\rho)D_K(\rho, \dot{\rho})C_2(\rho) & B_2(\rho)C_K(\rho, \dot{\rho}) \\ B_K(\rho)C_2(\rho) & A_K(\rho, \dot{\rho}) \end{bmatrix}, \\ B_{clp}(\rho, \dot{\rho}) &= \begin{bmatrix} B_{11}(\rho) & B_{12}(\rho) + B_2(\rho)D_K(\rho, \dot{\rho}) \\ 0 & B_K(\rho, \dot{\rho}) \end{bmatrix}, \\ C_{clp}(\rho, \dot{\rho}) &= \begin{bmatrix} C_{11}(\rho) & 0 \\ C_{12}(\rho) + D_K(\rho, \dot{\rho})C_2(\rho) & C_K(\rho, \dot{\rho}) \end{bmatrix}, \\ D_{clp}(\rho, \dot{\rho}) &= \begin{bmatrix} 0 & 0 \\ 0 & D_K(\rho, \dot{\rho}) \end{bmatrix}. \end{aligned} \tag{3.15}$$

In the following section, we define the quadratic LPV γ-performance problem using PDLF.

3.4 Parameter Dependent Output Feedback Problem and Controller Synthesis

In this section, we will define the parameter dependent output feedback problem. It is to determine if there exists a parameter dependent controller such that the analysis test described in the following lemma holds for the closed-loop system.

Defintion 3 *[15]* ***Output Feedback*** $\boldsymbol{\gamma}$***-Performance Problem*** *Given the open-loop LPV system* G_ρ *in (3.12), and the performance level* $\gamma > 0$*, the output feedback* γ *performance problem is solvable if there exists a parameter dependent controller* $(A_K, B_K, C_K, D_K) : \mathbf{R}^s \times \mathbf{R}^s \to (\mathbf{R}^{m\times m}, \mathbf{R}^{m\times n^y}, \mathbf{R}^{n_u \times m}, \mathbf{R}^{n_u\times n_y})$ *such that* $W(\rho) > 0$ *and*

$$\begin{bmatrix} A_{clp}^T(\rho)W(\rho) + W(\rho)A_{clp}(\rho) + \sum_{i=1}^{s} \pm(\dot{\rho}_i \frac{\partial W}{\partial \rho_i}) & W(\rho)B_{clp}(\rho) & \gamma^{-1}C_{clp}^T(\rho) \\ B_{clp}^T(\rho)W(\rho) & -I_{n_d} & \gamma^{-1}D_{clp}^T(\rho) \\ \gamma^{-1}C_{clp}(\rho) & \gamma^{-1}D_{clp}(\rho) & -I_{n_e} \end{bmatrix} < 0 \tag{3.16}$$

for all $\rho \in \mathcal{P}$ *and* $|\dot{\rho}| \leq \nu_i, i = 1, \ldots, s$*. The closed-loop matrices* $A_{clp}, B_{clp}, C_{clp}, D_{clp}$ *are defined in equation (3.15).*

The notation $\sum_{i=1}^{s} \pm(\cdot)$ in (3.16) indicates that all combinations of the parameter value of ρ should be included in the inequality. To be more specific, consider there are n parameters forming a polytope with $2n$ extreme points, such that there exists 2^n combinations of extreme values. We only need to check the extreme values for LMI synthesis, therefore this 3×3 inequality actually represents 2^n different inequalities which must be checked simultaneously.

Note that if we chose W to be a constant matrix and write the first term in (3.16) as $A_{clp}^T(\rho)W + WA_{clp}(\rho)$, inequality (3.16) is reduced to an SQLF problem which is discussed in [17, 31]. We can see that SQLF is actually a special case of PDLF. Because inequality (3.16) is not linear in the matrices A_K, B_K, C_K, D_k and W, it cannot easily be used to synthesised a controller K_ρ that solves the quadratic γ performance problem. However, it can be shown that by partitioning the Lyapunov matrices with a change of variables technique [52], we can transform the original problem into a system of LMIs which can be readily solved by convex optimisation.

Before proceeding to the theorem, we will define some notation. For all $W \in R^{(n+m)\times(n+m)}, W = W^T > 0$, partition W and W^{-1} as follows

$$W = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix}, \quad W^{-1} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix},$$

where $X_{11} \in R^{n\times n}, X_{11} = X_{11}^T > 0, Y_{11} \in Y_{11}^T > 0$ and $X_{12}, X_{22}, Y_{12}, Y_{22}$ are of appropriate dimensions. We also denote

$$\begin{bmatrix} D_{111\cdot}(\rho) \\ D_{112\cdot}(\rho) \end{bmatrix} := \left[\begin{array}{cc} D_{1111}(\rho) & D_{1112}(\rho) \\ \hline D_{1121}(\rho) & D_{1122}(\rho) \end{array}\right],$$

$$[D_{11\cdot 1}(\rho) \quad D_{11\cdot 1}(\rho)] := \left[\begin{array}{c|c} D_{1111}(\rho) & D_{1112}(\rho) \\ D_{1121}(\rho) & D_{1122}(\rho) \end{array}\right].$$

We now state the theorem for solving the quadratic LPV γ performance problem with SQLF given in [31].

Theorem 1 *Given $\mathcal{P}$, and the open-loop system in equation (3.2), the quadratic LPV γ performance problem is solvable if and only if there exist symmetric matrices $X_{11} \in R^{n\times n}$ and $Y_{11} \in R^{n\times n}$, such that for all $\rho \in \mathcal{P}$*

$$\begin{bmatrix} Y_{11}\hat{A}^T(\rho) + \hat{A}(\rho)Y_{11} - B_2(\rho)B_2^T(\rho) & Y_{11}C_{11}^T(\rho) & \gamma^{-1}\hat{B}_(\rho) \\ C_{11}(\rho)Y_{11} & -I_{e_1} & \gamma^{-1}D_{111\cdot}(\rho) \\ \gamma^{-1}\hat{B}^T(\rho) & \gamma^{-1}D_{111\cdot}^T(\rho) & -I_{n_d} \end{bmatrix} < 0$$

$$\begin{bmatrix} \tilde{A}^T(\rho)X_{11} + X_{11}\tilde{A}(\rho) - C_2^T(\rho)C_2(\rho) & X_{11}B_{11}(\rho) & \gamma^{-1}\tilde{C}_(\rho) \\ B_{11}(\rho)X_{11} & -I_{d_1} & \gamma^{-1}D_{11\cdot 1}^T(\rho) \\ \gamma^{-1}\tilde{C}(\rho) & \gamma^{-1}D_{11\cdot 1}(\rho) & -I_{n_e} \end{bmatrix} < 0 \tag{3.17}$$

$$\begin{bmatrix} X_{11} & \gamma^{-1}I_n \\ \gamma^{-1}I_n & Y_{11} \end{bmatrix} \geq 0$$

where

$$\hat{A}(\rho) := A(\rho) - B_2(\rho)C_{12}(\rho), \qquad \hat{B}(\rho) := B_1(\rho) - B_2(\rho)D_{112\cdot}(\rho)$$

$$\tilde{A}(\rho) := A(\rho) - B_{12}(\rho)C_2(\rho), \qquad \tilde{C}(\rho) := C_1(\rho) - D_{11\cdot 2}(\rho)C_2(\rho).$$

The formulas for an n-dimensional, strictly proper controller that solves the quadratic LPV γ-performance problem can be explicitly given as follows: Let $Q := (X_{11} - \gamma^{-2}Y_{11}^{-1})$, and define

$$F(\rho) := -(D_{12}^T(\rho)D_{12}(\rho))^{-1}(B^T(\rho)Y_{11}^{-1} + D_{12}^T(\rho)C_1(\rho)),$$

and

$$L(\rho) := -(X_{11}C^T(\rho) + B_1(\rho)D_{21}^T(\rho))(D_{21}D_{21}^T(\rho))^{-1}.$$

Let

$$\begin{aligned} H(\rho) := -[&Y_{11}^{-1}A_F(\rho) + A_F(\rho)^T Y_{11}^{-1} + C_F^T(\rho)C_F(\rho) \\ &+(Y_{11}^{-1}B_1(\rho) + C_F^T(\rho)D_{11}(\rho))(\gamma^{-2}I - D_{11}^T(\rho)D_{11}(\rho))^{-1}(B_1^T(\rho)Y_{11}^{-1} + D_{11}^T C_F(\rho)) \end{aligned}$$

where $A_F := A(\rho) + B_2(\rho)F(\rho)$, and $C_F := C_1(\rho) + D_{12}F(\rho)$, and

$$\begin{aligned} M(\rho) := &F^T(\rho)[B^T(\rho)Y_{11}^{-1} + D_{12}^T(C_1(\rho) + D_{12}F(\rho))] + H(\rho) \\ &+[\gamma^{-2}Q(-Q^{-1}X_{11}L(\rho)D_{21} - B_1) + F(\rho)^T D_{12}^T D_{11}(\rho)] \\ &*[\gamma^2 I - D_{11}^T(\rho)D_{11}(\rho)]^{-1}[B_1^T(\rho)Y_{11}^{-1} + D_{11}^T(C_1(\rho) + D_{12}F(\rho)]. \end{aligned}$$

Then the state space data for the controller K_ρ is defined as

$$\begin{aligned} A_K &:= A(\rho) + B_2(\rho)F(\rho) + Q^{-1}X_{11}L(\rho)C_2(\rho) - \gamma^{-2}Q^{-1}M(\rho) \\ B_K &:= -Q^{-1}X_{11}L(\rho) \\ C_K &:= F(\rho) \\ D_K &:= 0. \end{aligned} \tag{3.18}$$

Note that three LMIs in (3.17) trivially represent convex constraints on the matrices X_{11} and Y_{11}. The first two LMIs depend continuously on $\rho \in \mathcal{P}$, and the solution to the quadratic LPV γ performance problem can be regarded as a convex feasibility problem with infinite number of constraints. Now we will state the γ performance output feedback synthesis problem using PDLF, with the parameter dependent quadratic stability and performance results derived in section 3.2.

Theorem 2 *[15] Given a compact set $\mathcal{P}$, the performance level γ and the LPV system in (3.2), the parameter dependent γ-performance problem is solvable if and only if there exist matrix functions $X_{11} \in \mathcal{C}^1(\mathbf{R}^s, \mathcal{S}^{n\times n})$ and $Y_{11} \in \mathcal{C}^1(\mathbf{R}^s, \mathcal{S}^{n\times n})$, such that for all $\rho \in \mathcal{P}, X(\rho), Y(\rho) > 0$, and*

$$\begin{bmatrix} Y_{11}(\rho)\hat{A}^T(\rho) + \hat{A}(\rho)Y_{11}(\rho) - \sum_{i=1}^{s} \dot{\rho}_i \frac{\partial Y(\rho)}{\partial \rho_i} - B_2(\rho)B_2^T(\rho) & Y_{11}(\rho)C_{11}^T(\rho) & \gamma^{-1}\hat{B}_(\rho) \\ C_{11}(\rho)Y_{11}(\rho) & -I_{e_1} & \gamma^{-1}D_{111\cdot}(\rho) \\ \gamma^{-1}\hat{B}^T(\rho) & \gamma^{-1}D_{111\cdot}^T(\rho) & -I_{n_d} \end{bmatrix} < 0$$

$$\begin{bmatrix} \tilde{A}^T(\rho)X_{11}(\rho) + X_{11}(\rho)\tilde{A}(\rho) + \sum_{i=1}^{s} \dot{\rho}_i \frac{\partial X(\rho)}{\partial \rho_i} - C_2^T(\rho)C_2(\rho) & X_{11}(\rho)B_{11}(\rho) & \gamma^{-1}\tilde{C}_(\rho) \\ B_{11}(\rho)X_{11}(\rho) & -I_{d_1} & \gamma^{-1}D_{11\cdot 1}^T(\rho) \\ \gamma^{-1}\tilde{C}(\rho) & \gamma^{-1}D_{11\cdot 1}(\rho) & -I_{n_e} \end{bmatrix} < 0$$

$$\begin{bmatrix} X_{11}(\rho) & \gamma^{-1}I_n \\ \gamma^{-1}I_n & Y_{11}(\rho) \end{bmatrix} \geq 0 \qquad (3.19)$$

A controller that solves the parameter dependent γ performance output feedback problem, has the same structure as that given in (3.18), but with X_{11} and Y_{11} now parameter dependent and

$$\begin{aligned} H(\rho) := -\Big[&X_{11}^{-1}(\rho)A_F(\rho) + A_F^T(\rho)X_{11}^{-1}(\rho) + C_F^T(\rho)C_F(\rho) \\ &+(X_{11}^{-1}(\rho)B_1(\rho) + C_F^T(\rho)D_{11}(\rho)) \\ &*(\gamma^2 I - D_{11}^T(\rho)D_{11}(\rho))^{-1}(B_1^T(\rho)X_{11}^{-1}(\rho) + D_{11}^T C_F(\rho))\Big] \end{aligned}$$

A controller K_ρ that solves the LPV γ performance problem, will guarantee exponential stability for arbitrarily fast variation of ρ in the case of SQLF. But in reality, the parameter rates of variation do not vary arbitrarily fast. Therefore, by employing PDLF, we take the bounds of the variation rates into account during the synthesis and it may possible reduce the conservatism, but with a cost of greater computational complexity.

The synthesis problem is typically solved by gridding the set $\mathcal{P}$, and choosing the grid points to describe the operating range of the physical plant. If the plant model is quasi-LPV, the scheduling parameters will include some of the states of the plant. There might be some grid points that correspond to values of ρ that cannot be achieved in the applications. For example, considering an aircraft scheduled on velocity and angle of attack, it is impossible for the aircraft to operate at high angles of attack while maintaining high airspeeds. Thus, grid points at high angles of attack and high airspeeds can be excluded from the parameter set which will probably lead to a smaller value of γ. The number of LMI constraints will also depend on the number of the grid points. For example, in a SQLF synthesis, if the scheduling parameter $\rho = [\alpha_1, \alpha_2]^T$ and the range of α_1, α_2 is spanned by $n_{\alpha_1}, n_{\alpha_2}$ grid points respectively, then the number of LMI constraints for SQLF synthesis is

$$N_{SQLF} = 2n_{\alpha_1}n_{\alpha_2} + 1$$

and the number of the decision variables is $n(n+1)/2+1$, where n is the number of the states.

For the synthesis using PDLF, the problem will become more complicated. The number of the LMI constraints is

$$N_{PDLF} = (2^s + 1)n_{\alpha_1}n_{\alpha_2}$$

where 2^s is the number of vertices in the polytope $\mathcal{V}$, which characterises the permissible parameter space of $\dot{\rho}$. The number of decision variables depends on the parameterisations of X_{11} and Y_{11}. Without loss of the generality, $X(\rho)$ and $Y(\rho)$ can be given as in [18]

$$X_{11}(\rho) = \sum_{i}^{n_x} f_i(\rho)X_i, \qquad Y_{11}(\rho) = \sum_{i}^{n_y} g_i(\rho)Y_i,$$

where $f_i(\rho)$ and $g_i(\rho)$ are continuous and differentiable functions of ρ. Here we assume that the number of basis functions of X_{11} and Y_{11} are n_x and n_y respectively, and the number of decision variables is given by $n(n+1)(n_x+n_y)/2+1$. Note that the number of LMIs increases exponentially with the number of scheduling parameters even without considering the implicit dependence of the number of grid point on the number of parameters. Hence it is suggested that the synthesis method will be most efficient for problems involving only a few (e.g., one or two) parameters.

The synthesised controller K_ρ comes with the following guarantee about the LPV closed loop

$$\sup_{\rho \in \mathcal{F}_{\mathcal{P}}} \sup_{\|d\|_2 \neq 0, d \in L_2} \frac{\|e\|_2}{\|d\|_2} < \gamma < \infty. \tag{3.20}$$

As the plant (3.2) and K_ρ are linear, from the small gain theorem we know that the interconnection systems of the perturbed plant and K_ρ will be well-posed and internally stable for any nonlinear, time-varying perturbation $\Delta : e \mapsto d$, with an incremental gain smaller than or equal to $1/\gamma$. It can be deduced from the bounded real lemma that γ derived from the quadratic γ performance synthesis is an upper bound for those achieved from evaluating the LTI stability margin. Hence, it is possible to calibrate γ to gain better robustness. We will discuss this in detail in the next chapter.

It is assumed throughout this thesis that the parameter ρ is known in real-time. This is not completely true because of sensor noise. The effects of the noise on the closed-loop robust stability is difficult to analyse because this type of sensor noise is not considered in the quadratic γ performance design framework. If the plant is very nonlinear, then it is highly possible that large bias in the measurement could cause the closed-loop system to become unstable. Also time delays in sensing the scheduled variables can result in an unstable closed-loop system. Again it is difficult to analyse the robustness of the closed-loop system under the effects of time delay.

Chapter 4

Robust control system design for LPV systems

In this chapter, we will develop two design methods using parameter dependent Lyapunov functions (PDLFs). First we will introduce a two-degree-of-freedom (2DoF) mixed sensitivity design. The conventional PDLF design for an LPV system uses a uniform bound γ on the L_2 norm, the resulting controller has to be synthesised for the worst case scenario, and the overall robust performance index of the controller will suffer because of a limited number of operating points. The idea of a parameter dependent performance index was first used in the design of an LPV flight controller for an F-16 aircraft [34]. We will also use a parameter-dependent performance index in our 2DoF/PDLF synthesis and demonstrate that the design process can be less computationally expensive and more intuitive with a simple parameterisation.

In the second part of the chapter, we will focus on a loop shaping design approach which also uses a PDLF method. We first follow the method in [43] to derive the left coprime factorisation of the LPV plant. Then we present the solvability conditions for the γ performance output feedback problem. The left coprime factorisation structure can be fitted nicely into Wu's method with some modification of the original assumptions. We will show that the PDLF method can also reduce conservatism effectively in the loop shaping design when compared to the SQLF approach.

4.1 A two-degree-of-freedom PDLF/Design Method

4.1.1 Parameter Dependent Performance

In the literature on parameter dependent LPV system synthesis, the performance level is usually restricted to be constant. The performance γ represents the upper L_2 norm bound from input e to output d as

$$\gamma > \sup_{\gamma \text{ stabilizing}} \frac{\|e\|_2}{\|d\|_2}.$$

In this way, the system has overall γ performance corresponding to the worst case scenario, e.g., for an aircraft at high angle of attack or operating at the edge of its flight envelope, the performance of the system operating at its benign operating points has to be sacrificed.

An LPV system represents the system dynamics as functions of a varying parameter trajectory, and at each “frozen” operating point, the system has different levels of tolerance to perturbations and disturbances. It is appealing therefore to consider γ varying with different operating points. In [34], the parameter varying γ performance was first adopted in an LPV synthesis for an F-16 aircraft. The methodology that was implemented in that paper generalised γ as an integral quadratic constraints and used an iterative procedure to optimise the objective function

$$\lambda(\gamma) := \int_{\mathcal{P}} w(\rho)\gamma(\rho)d\rho,$$

where $w(\rho)$ is the weighting function that is determined at each frozen point of the parameter trajectories. First the objective function $\lambda(\gamma)$ is minimised subject to the γ performance LMI constraints. In the absence of priori information on γ, an initial guess for the weighting function has to be made and a pre-specified value γ_{ok} is used to setup the stopping rule. The procedure can be summarised as follows:

1. Solve the minimisation problem subject to the LMI constraints.

2. If $\max\gamma(\rho) \leq \gamma_{\text{ok}}$, then stop.

3. Otherwise, set the next weighting function $w_{j+1} = (\rho(j))^j$ and repeat.

There are several potential problems with this procedure. First of all, the choice of the initial weighting function will be critical to determine the times that are required to repeat the procedure. A variable γ_i has to be assigned to each operating point, and this greatly increases the number of LMI variables to be solved. The pre-specified optimal γ_{ok} needs to be "guessed" from the LMI synthesis results with constant γ, and it might cause the optimisation results to become sub-optimal. The updating rule for the weighting function is arbitrary, therefore the LMIs could become infeasible. Finally, the iterative procedure is time consuming especially if the LMI synthesis problem is multi-dimensional. The method used in our 2DoF/PDLF design will be more intuitive and straightforward. With a simple parameterisation, we can achieve the same result with much better efficiency.

Defintion 4 *[35] Consider the LPV plant $G(\rho)$ satisfying assumptions A1-A3 in §4.3, and a parameter trajectory ρ, such that its derivative $\dot{\rho}$ exists and lies in the polytope $\{\mathcal{V}_\rho\}_{i=1}^{s}$ with s vertices. The parameter dependent performance output-feedback problem is solvable for the closed-loop system in (3.15) if there exist a continuous differentiable matrix function $W(\rho) : R^s \rightarrow \mathcal{S}^{(n+m)\times(n+m)}$, $W(\rho) > 0$ and*

$$\begin{bmatrix} A_{clp}^T(\rho)W(\rho) + W(\rho)A(\rho)_{clp} + \sum_{i=1}^{s} \dot{\rho}_i \dfrac{\partial W(\rho)}{\partial \rho_i} & W(\rho)B_{clp}(\rho) & C_{clp}^T(\rho) \\ B_{clp}^T(\rho)W(\rho) & -\gamma(\rho)I_{n_d} & D_{clp}^T(\rho) \\ C_{clp}(\rho) & D_{clp}(\rho) & -\gamma(\rho)I_{n_e} \end{bmatrix} < 0. \quad (4.1)$$

The closed-loop matrices $A_{clp}(\rho), B_{clp}(\rho), C_{clp}(\rho)$ and $D_{clp}(\rho)$ are as defined in (3.15). If the LPV synthesis problem is solvable, then the induced-L_2 norm is less than $\gamma(\rho)$, which is a continuous function of ρ.

This is the Bounded Real Lemma reformulated with parameter dependent performance using quadratic parameter dependent Lyapunov functions. The following theorem gives the sufficient conditions for the existence of an output-feedback controller that gives the closed-loop has a parameter dependent γ performance level. For simplicity, we assume that $D_{11} = 0$ from now on.

Theorem 3 *[35] Given the LPV plant as in (3.13), there is an output feedback controller such that the closed loop system in (3.14) has a parameter dependent quadratic performance level of $\gamma(\rho)$, iff there exists parameter dependent Lyapunov matrices $X(\rho) > 0$, $Y(\rho) > 0$ satisfying the following inequalities:*

$$\begin{bmatrix} \hat{A}(\rho)Y(\rho) + Y(\rho)\hat{A}^T(\rho) - \sum_{i=1}^{s} \dot{\rho}_i \dfrac{\partial Y(\rho)}{\partial \rho_i} - \gamma(\rho)B_2(\rho)B_2^T(\rho) & Y(\rho)C_{11}^T(\rho) & B_1(\rho) \\ C_{11}(\rho)Y(\rho) & -\gamma(\rho)I_{n_{e_1}} & 0 \\ B_1^T(\rho) & 0 & -\gamma(\rho)I_{n_d} \end{bmatrix} < 0 \tag{4.2}$$

$$\begin{bmatrix} \tilde{A}^T(\rho)Y(\rho) + X(\rho)\tilde{A}(\rho) + \sum_{i=1}^{s} \dot{\rho}_i \dfrac{\partial X(\rho)}{\partial \rho_i} - \gamma(\rho)C_2^T(\rho)C_2(\rho) & X(\rho)B_{11}(\rho)^T & C_1^T(\rho) \\ B_{11}(\rho)^T X(\rho) & -\gamma(\rho)I_{n_{d1}} & 0 \\ C_1(\rho) & 0 & -\gamma(\rho)I_{n_e} \end{bmatrix} < 0 \tag{4.3}$$

$$\begin{bmatrix} Y(\rho) & I_n \\ I_n & X(\rho) \end{bmatrix} > 0 \tag{4.4}$$

at all $(\rho, \dot{\rho})$ where

$$\tilde{A}(\rho) = A(\rho) - B_2(\rho)C_{12}(\rho), \qquad \hat{A}(\rho) = A(\rho) - B_1(\rho)C_2(\rho).$$

The proof of the theorem can be found in [15, 23], for the case when the performance index γ is constant. In Theorem 3, we let γ depend on parameter trajectory ρ.

To implement the idea in the robust LMI synthesis, the dependence of $\gamma(\rho)$ on ρ has to be decided. We use the same principal of heuristically choosing the parameterisation for the Lyapunov variables X, Y: the dependence of $\gamma(\rho)$ on ρ will be similar to the system dynamics that depend on the scheduling parameters [18, 23]:

$$\gamma(\rho) = \gamma_0 + \sum_{i=1}^{N} \theta_i(\rho)\gamma_i. \tag{4.5}$$

This parameterisation implies that γ is intuitively related to the dynamics of the LPV system in terms of the scheduling parameters. Compared to the iterative procedure,

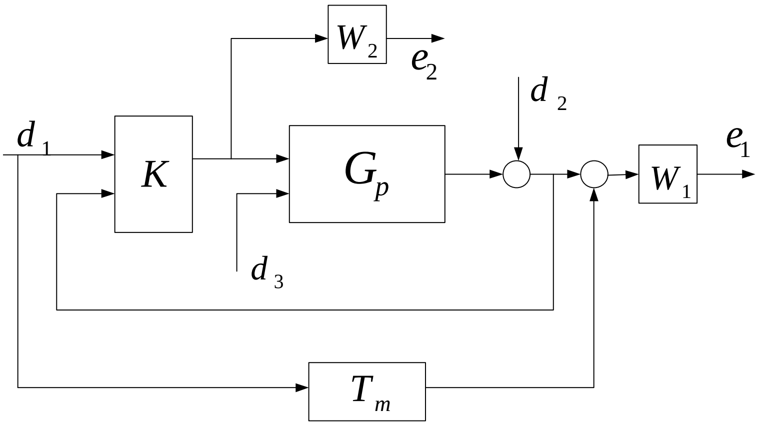


Figure 4.1: Two Degree of Freedom control structure

this method only requires the LMIs to be solved once, and the number of γ variables, which is substantially less than the method in [34], is determined by the parameterisation of the polynomial. The necessary and sufficient conditions in Theorem 3 are a systems of LMIs with functional variables X, Y and ρ. Therefore, the convex optimisation problem can be stated formally as

$$\text{minimise} \quad \gamma(\rho) \ \text{ subject to } \ (4.2) - (4.4). \tag{4.6}$$

4.1.2 The Two-degree-of-freedom LPV control and Design Procedure

In this section we will describe the implementation of a mixed-sensitivity model matching 2DoF design with parameter dependent γ performance. The background to 2DoF design can be found in [45, 53]. A 2DoF design structure can be characterised as in Figure 4.1. In this system, d_1 is the reference input, d_2 and d_3 are perturbation inputs, e_1 are the performance outputs, e_2 are the weighted control outputs.

We can represent the input-output relationship by the following equation

$$\begin{bmatrix} e_1 \\ e_2 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -W_1T_m & W_1 & W_1G(\rho) & W_1G(\rho) \\ 0 & 0 & 0 & W_2 \\ I & 0 & 0 & 0 \\ 0 & I & G & G \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ u \end{bmatrix}, \tag{4.7}$$

such that the transfer function from e to d can be written as

$$T_{ed}(\rho) = \begin{bmatrix} W_1(S_oG(\rho)K - M) & W_1S_o & W_1SoG(\rho) \\ W_2S_iK_i & W_2KS_o & W_2K_2S_oG(\rho) \end{bmatrix}, \tag{4.8}$$

where S_i and S_o denote the input and output sensitivity functions respectively. The functions in $T_{ed}(\rho)$ have the following meanings:

$W_1(S_oG(\rho)K - T_m)$	Weighted difference between ideal and actual model
W_1S_o	Weighted output sensitivity
W_2S_iK	Weighted control effort due to reference inputs
W_2KS_o	Weighted control effort due to output disturbances

For simplicity we omit the dependency on ρ and let the state space representations of the plant and weights in Figure 4.1 be

$$\begin{gathered} G_P = \left[\begin{array}{c|cc} A_p & B_{p1} & B_{p2} \\ \hline C_p & D_{p1} & D_{p2} \end{array}\right] \qquad T_m = \left[\begin{array}{c|c} A_m & B_m \\ \hline C_m & D_m \end{array}\right] \\ W_1 = \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right] \qquad W_2 = \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right], \end{gathered} \tag{4.9}$$

The state space form of generalised plant can be constructed as

$$\left[\begin{array}{c} \dot{x} \\ \hline e \\ y \end{array}\right] = \left[\begin{array}{c|cc} A_g & B_{g1} & B_{g2} \\ \hline C_{g1} & D_{g11} & D_{g12} \\ C_{g2} & D_{g21} & D_{g22} \end{array}\right] \left[\begin{array}{c} x \\ \hline d \\ u \end{array}\right] \tag{4.10}$$

With the state-space matrices in (4.10), we can solve the synthesis problem using LMIs (4.2)–(4.4) in Theorem 3. The objective of the synthesis is to minimise $\|T_{ed}(\rho)\|_2$ to mitigate the effects of the output disturbances on the tracking errors and control effort, while guaranteeing stability of the closed-loop system.

Lemma 11 *[34] Given the closed-loop system in (3.13), if there exists $X(\rho), Y(\rho)$ and $\gamma(\rho)$ for LMIs (4.2)-(4.4) at all $(\rho, \dot{\rho})$ for which $\rho \in \mathcal{P}$ and $\dot{\rho} \in (\mathcal{V}_\rho)$, then we can construct an output feedback controller as follows: let $N, M \in \mathcal{C}^1(R^s, R_+^{n\times n})$ be matrix functions that solve the factorisation problem*

$$I - XY = NM^T. \tag{4.11}$$

Define the short-hand notation

$$F = -(\gamma B_2^T Y^{-1} + D_{12}^T C_1)$$

and

$$H = -(\gamma X^{-1} C_2^T + B_1 D_{21}^T),$$

Then one n-dimensional, strictly proper controller that solves the feedback problem is given by

$$\begin{aligned} A_K &:= -N^{-1}\{A^T + X(A + B_2F + HC_2)Y \\ &\quad +C_1^T(C_1 + D_{12}F)Y/\gamma + X(B_1 + HD_{21})B_1^T/\gamma \\ &\quad +\dot{X}Y + \dot{N}M^T\}M^{-T}, \\ B_K &:= N^{-1}XH, \\ C_K &:= FYM^{-T}, \\ D_K &:= 0. \end{aligned} \tag{4.12}$$

The proof of this controller formula can be found in [15]. The synthesised controller $K(\rho)$ comes with the following guarantee about the LPV closed-loop system:

$$\sup_{\rho\in\mathcal{D}} \frac{\|e\|_2}{\|d\|_2} < \gamma(\rho) < \infty. \tag{4.13}$$

As weighted plant (4.10) and $K(\rho)$ are linear, the small-gain theorem and (4.13) guarantee that the closed-loop interconnected system of the perturbed weighted plant and $K(\rho)$ will be well-posed and internally stable for any nonlinear, possibly time-varying, infinite-dimensional perturbation $\Delta : e \mapsto d$, with an incremental gain smaller than or equal to $\gamma(\rho)^{-1}$. If γ is constant, from the bounded real lemma and (4.13), the stability margin evaluated at each "frozen" point is an upper bound of γ^{-1}. From the design example in the chapter 6, it can be shown that the performance index evaluated at the most "troublesome" operating points will be approximately equal to γ if it was chosen to be constant.

The characterisation of Theorem 3 involves solutions of a convex but infinite dimensional and infinitely constrained problem, because we allow a general parameter dependence in the systems. In order to make the problem tractable, we can grid the parameter space $\mathcal{P}$ and solve the convex optimisation problem at each grid point. The derivative $\dot{\rho}$ appears linearly in the LMIs, so we only have to check the vertices of the polytope $\mathcal{V}_\rho$. The functional dependence of the Lyapunov matrices X and Y can be selected similarly as discussed in the previous section. Given the set of differentiable functions $\theta_i(\rho)$, we can parameterize X and Y as

$$X(\rho) = X_0 + \sum_{i=1}^{n} \theta(\rho) X_i, \quad Y(\rho) = Y_0 + \sum_{i=1}^{n} \theta(\rho) Y_i. \tag{4.14}$$

With the functional dependence fixed, the parameterisations of X and Y decide the number of the decision variables in the LMI problems (4.2)–(4.4).

From (4.12) we can see that in general formula of A_K depends explicitly not only on the real-time measurements of ρ, but also its time derivative $\dot{\rho}$ via $(\dot{X}, \dot{N})$. It is undesirable if there are sensor noise and inaccuracies in the parameter measurements. A simple approach to tackle this problem has been proposed in [33], where it is shown shown that a controller (which does not depend on $\dot{\rho}$) can be synthesised by choosing either X or Y to be constant. To be more specific, let us differentiate both sides of the equation in (4.11) to get the identity

$$\dot{X}Y + \dot{N}M^T = -X\dot{Y} - N\dot{M}^T.$$

If X is chosen to be constant, and N, M are factorised as

$$(N, M) = (I, I - YX),$$

then the term $\dot{X}Y + \dot{N}M^T$ in A_K is equal to zero. Alternatively, we can eliminate the derivation term by choosing Y to be constant, and factorising N, M as

$$(N, M) = (I - YX, I).$$

Since $\rho \subset R^s$, it will require approximately L^s points to grid $\mathcal{P}$ with approximately L points in each dimension. If the LMIs are satisfied only at the grid point of $\mathcal{P}$, and the synthesis conditions may not be satisfied at some parameter values that lie between the grid points. One generally needs to check the computed solution against a sufficiently fine grid; if the check fails, repeat the synthesis procedure using more grid points.

A general parameter dependent performance 2DoF output feedback design procedure can be summarised as follows:

1. Define a grid for the value set of ρ.

2. Select the matching model T_m and weighting functions W_1, W_2.

3. Use the loop transformation to convert the closed-loop plant into a simplified form.

4. Select the parameterisations for $X(\rho), Y(\rho)$ and $\gamma(\rho)$ according to the parameter dependence of the original plant.

5. Minimise $\gamma(\rho)$ subject to the LMI constraints.

6. If step 5 fails, increase the density of the grid and return to step 2.

7. Construct a controller with the feasible solutions from the LMI synthesis.

4.2 A Loop Shaping/PDLF Design Method

In this section, we extend the loop shaping design method for LPV systems in [42] using a parameter dependent Lypunov function (PDLF) [46, 47] and Wu's framework [15, 23]. We first follow the method in [43] to derive a left coprime factorisation of the LPV plant. Then we present the solvability conditions for the γ-performance loop shaping problem. The left coprime factorisation structure can be fitted nicely into Wu's method with some modification of the original assumptions. A PDLF approach incorporates the bounded parameter rate of variation into the synthesis, and therefore offers a less conservative solution to the LMI synthesis. The synthesis procedure can be easily implemented in the Matlab LMI toolbox [63]. The reconstruction formulas of the controller are also given explicitly, and do not require any additional LMIs to be solved.

4.2.1 Left Coprime Factorisation of LPV Systems

We consider a minimal realisation of a compensated (shaped) plant $G_s(\rho)$ as

$$G_s(\rho) = \left[\begin{array}{c|c} A(\rho) & B \\ \hline C & 0 \end{array}\right], \tag{4.15}$$

where the pairs $(A(\rho), B)$ and $(A(\rho), C)$ are controllable and observable with $A(\rho) \in R^{n\times n}$, $B(\rho) \in R^{n\times n_u}$ and $C(\rho) \in R^{n_y\times n}$. $D = 0$ implies the plant is strictly proper. If the system does not have constant B and C matrix, we can use pre-filtering techniques that are introduced in [17] to incorporate all the uncertainties in the A matrix and leave B and C constant.

Considering coprime factor uncertainties of the shaped plant, we have the normalised left coprime factorisation of $G_s(\rho) = \tilde{M}(\rho)^{-1}\tilde{N}(\rho)$, where a state space realisation of $(\tilde{M}(\rho), \tilde{N}(\rho))$ is given in [51] as

$$(\tilde{M}(\rho)\ \tilde{N}(\rho)) = \left[\begin{array}{c|cc} A(\rho)+LC & L & B(\rho) \\ \hline C(\rho) & I & 0 \end{array}\right], \tag{4.16}$$

where

$$L = -P^{-1}C^T \tag{4.17}$$

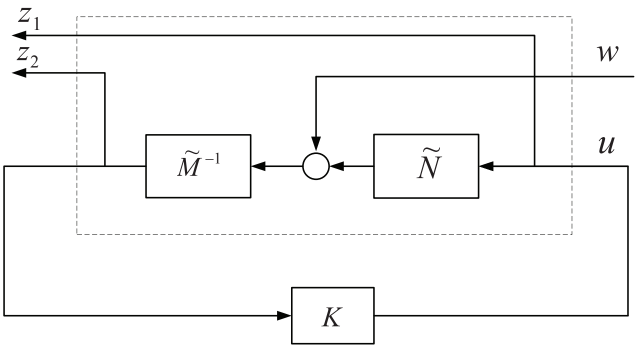


Figure 4.2: H_∞ loop shaping interconnection

is a stabilising observer gain. Here we assumes that the plant is strictly proper ($D = 0$), therefore R in (2.3) is equal to identity.

Finding a normalised left coprime factorisation for a parameter dependent system can be reduced to a special H_2 filtering problem. Here we will revisit the method.

Theorem 4 *There exist positive define $P = P^T$ and $Z = Z^T$ that solve a left coprime factorisation problem if the following optimisation problem is satisfied*

$$\begin{aligned} min \quad trace(Z), \\ \begin{bmatrix} PA(\rho) + A(\rho)^T P - C^T C & PB \\ B^T P & -I \end{bmatrix} < 0, \\ \begin{bmatrix} Z & I \\ I & P \end{bmatrix} > 0. \end{aligned} \tag{4.18}$$

Theorem 4 determines the left coprime factorisation of the parameter dependent system and a proof can be found in [43] . The conditions are numerically tractable as they are in LMI form. In the normalised coprime factor robust stabilisation framework, a parameter dependent controller is synthesised by satisfying

$$\inf_{K(\rho) \text{ stabilizing}} \left\| \begin{bmatrix} K(\rho) \\ I \end{bmatrix} (I + G_s(\rho)K(\rho))^{-1}\tilde{M}^{-1} \right\|_\infty < \gamma. \tag{4.19}$$

As we discussed in chapter 2, the pre- and post-compensators are chosen such that the shaped plant $G_s(\rho)$ has desired open loop shape and no hidden modes to ensure internal stability of the closed-loop system. The generalised plant corresponding to the loop shaping procedure in Figure 4.2 is given by

$$G_s(\rho) := \left[\begin{array}{c|c|c} A(\rho) & -L & B(\rho) \\ \hline 0 & 0 & I \\ C(\rho) & I & 0 \\ \hline C(\rho) & I & 0 \end{array}\right] \tag{4.20}$$

4.2.2 A Loop Shaping Design Using a Parameter Dependent Lyapunov Function

Change of assumptions

The parameter dependent Lyapunov function method in [23] requires the system to take on the structure in Eq.(3.13) and to satisfy assumptions A1–A3 in §3.3. The interconnection of (4.20) can be generalised as

$$G_s(\rho) := \left[\begin{array}{c|c|c} A(\rho) & B_1(\rho) & B_2(\rho) \\ \hline C_{11}(\rho) & D_{111}(\rho) & I_{n_u} \\ C_{11}(\rho) & D_{112}(\rho) & 0 \\ \hline C_2(\rho) & I_{n_y} & 0 \end{array}\right], \tag{4.21}$$

where

$$B_1(\rho) = -L, \quad B_2(\rho) = B(\rho), \quad \begin{bmatrix} C_{11}(\rho) \\ C_{12}(\rho) \end{bmatrix} = \begin{bmatrix} 0 \\ C(\rho) \end{bmatrix},$$

$$C_2(\rho) = C(\rho), \quad \begin{bmatrix} D_{111}(\rho) \\ D_{112}(\rho) \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

Observing the generalised loop shaping interconnection structure of (4.21), A1 is satisfied. Also note that to satisfy assumption A2, we only need to swap the second row of (4.20) with the third row. However, $D_{21} = I_{n_y}$ does not satisfy assumption A3, and therefore we have to modify the theorem to accommodate a change in the assumptions.

Remark 1 *It is suggested that we perform a norm-preserving loop transformation in §2.4 to obtain the generalized form in (3.13). Interestingly, in this case, the result of the loop transformation is exactly the same as swapping the second row of (4.20) with the third row and letting* $D_{21} = I_{n_y}$.

Theorem 5 *If the state space interconnection is given by (4.21), there is an output feedback controller such that the closed-loop system has quadratic performance of level* γ*, iff there exist parameter dependent Lyapunov matrices* $X(\rho) > 0$, $Y(\rho) > 0$ *which satisfy the following inequalities:*

$$\begin{bmatrix} \hat{A}(\rho)Y(\rho) + Y(\rho)\hat{A}^T(\rho) - \sum_{i=1}^{s} \dot{\rho}_i \frac{\partial Y(\rho)}{\partial \rho_i} - \gamma B_2(\rho)B_2^T(\rho) & Y(\rho)C_{11}^T(\rho) & B_1(\rho) \\ C_{11}(\rho)Y(\rho) & -\gamma I_{n_{e_1}} & 0 \\ B_1^T(\rho) & 0 & -\gamma I_{n_d} \end{bmatrix} < 0 \tag{4.22}$$

$$\begin{bmatrix} \tilde{A}^T(\rho)X(\rho) + X(\rho)\tilde{A}(\rho) + \sum_{i=1}^{s} \dot{\rho}_i \frac{\partial X(\rho)}{\partial \rho_i} - \gamma C_2^T(\rho)C_2(\rho) & C_1^T(\rho) \\ C_1(\rho) & -\gamma I_{n_{e_2}} \end{bmatrix} < 0 \tag{4.23}$$

$$\begin{bmatrix} Y(\rho) & I_n \\ I_n & X(\rho) \end{bmatrix} < 0 \tag{4.24}$$

at all $(\rho, \dot{\rho})$ *where*

$$\tilde{A}(\rho) = A(\rho) - B_2(\rho)C_{12}(\rho), \qquad \hat{A}(\rho) = A(\rho) - B_1(\rho)C_2(\rho).$$

Because we modify the original theorem due to the change in the assumptions, we give the following proof to justify that Theorem 5 still hold with the change of assumptions:

Proof 2 *Consider a closed-loop interconnection for the LPV plant in (4.21) as*

$$\begin{bmatrix} \dot{x}_{cl}(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} A_{cl}(\rho(t), \dot{\rho}(t)) & B_{cl}(\rho(t), \dot{\rho}(t)) \\ C_{cl}(\rho(t), \dot{\rho}(t)) & D_{cl}(\rho(t), \dot{\rho}(t)) \end{bmatrix} \begin{bmatrix} x_{cl}(t) \\ d(t) \end{bmatrix}, \tag{4.25}$$

where

$$A_{cl}(\rho,\dot\rho) = \begin{bmatrix} A(\rho) + B_2(\rho)D_K(\rho,\dot\rho)C_2(\rho) & B_2(\rho)C_K(\rho,\dot\rho) \\ B_K(\rho)C_2(\rho) & A_K(\rho,\dot\rho) \end{bmatrix},$$

$$B_{cl}(\rho,\dot\rho) = \begin{bmatrix} B_2(\rho) + B_2(\rho)D_K(\rho,\dot\rho) \\ B_K(\rho,\dot\rho) \end{bmatrix}. \tag{4.26}$$

$$C_{cl}(\rho,\dot\rho) = \begin{bmatrix} C_{11}(\rho) & 0 \\ C_{12}(\rho) + D_K(\rho,\dot\rho)C_2(\rho) & C_K(\rho,\dot\rho) \end{bmatrix},$$

$$D_{cl}(\rho,\dot\rho) = \begin{bmatrix} 0 \\ D_K(\rho,\dot\rho) \end{bmatrix}.$$

Let W be the continuous differentiable parameter dependent Lyapunov function that satisfies the Bounded Real Lemma in (4.1) for closed-loop system (4.25). Using a change of variables that was proposed in [52] We partition W and W^{-1} as

$$W = \begin{bmatrix} X & N \\ N^T & Q^{-1} \end{bmatrix} \quad W^{-1} = \begin{bmatrix} Y & M \\ M^T & * \end{bmatrix} \tag{4.27}$$

Using lemmas 2 and 5 in §2.1, $W > 0$ implies that for all $\rho \in \mathcal{P}$

$$\begin{bmatrix} X(\rho) & I \\ I & Y(\rho) \end{bmatrix} > 0.$$

Using the closed-loop state space matrices in (4.26), we can write the LMI in (4.1) as

$$R(\rho) + U(\rho)K(\rho)V^T(\rho) + V(\rho)K(\rho)U^T(\rho) < 0, \tag{4.28}$$

where

$$R := \begin{bmatrix} \begin{bmatrix} A^T & 0 \\ 0 & 0 \end{bmatrix} W + W \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} + \sum_{i=1}^{s} \dot\rho_i \frac{\partial W(\rho)}{\partial \rho_i} & W \begin{bmatrix} B_1^T \\ 0 \end{bmatrix} & \begin{bmatrix} C_{11}^T & C_{12}^T \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} B_1^T & 0 \end{bmatrix} W & -\gamma I & 0 \\ \begin{bmatrix} C_{11} & 0 \\ C_{12} & 0 \end{bmatrix} & 0 & -\gamma I \end{bmatrix}$$

$$U = \begin{bmatrix} W\begin{bmatrix} B_2 & 0 \\ 0 & I \end{bmatrix} \\ 0 \quad 0 \\ 0 \quad 0 \\ I \quad 0 \end{bmatrix} \qquad V = \begin{bmatrix} C_2^T & 0 \\ 0 & I \\ I & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

and $K = \begin{bmatrix} D_K(\rho) & C_K(\rho) \\ B_K(\rho) & A_K(\rho) \end{bmatrix}$ *The orthonormal basis for* U *and* V *can be defined as*

$$U_\perp = \begin{bmatrix} Y & 0 & 0 \\ M^T & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ -B_2^T & 0 & 0 \end{bmatrix} \qquad V_\perp = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ -C_2^T & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

and thus for all $\rho \in \mathcal{P}$ *,* $U_\perp^T U = 0$ *and* $V_\perp^T V = 0$ *. Note that* $[U \quad U_\perp^T]$ *and* $[V \quad V_\perp^T]$ *have full rank. Since both* $U_\perp$ *and* $V_\perp$ *have full column rank, it is clear that using Lemma 3, for all* $\rho \in \mathcal{P}$*, we have*

$$U_\perp^T(\rho)R(\rho)U_\perp(\rho) < 0 \;\; \textit{and} \;\; V_\perp^T(\rho)R(\rho)V_\perp(\rho) < 0$$

which can be verified to be identical to the LMIs (4.22) and (4.23) respectively by simple algebra.

Development of the loop shaping/PDLF method

In this section, we will present a useful theorem for loop shaping/PDLF synthesis. The left coprime factorisation of the LPV system that is used in the method comes directly from Theorem 4. The sufficient conditions of this theorem are derived from the H_∞ interconnection as in Figure 4.2 and Theorem 5.

Theorem 6 *Given the realisation of a left coprime factorization of the plant in Eq.(4.20), and* L *as in (4.17), there is a loop shaping controller* $K(\rho)$ *that satisfies (4.19), if there exist positive define matrices* $X(\rho)$ *and* $Y(\rho)$ *and scalar* γ *such* *that*

$$\begin{bmatrix} A(\rho)Y(\rho)+Y(\rho)A^T(\rho)-\sum_{i=1}^{s}\dot{\rho}_i\frac{\partial Y(\rho)}{\partial\rho_i}-\gamma B(\rho)B^T(\rho) & Y(\rho)C^T(\rho) & -L \\ C(\rho)Y(\rho) & -\gamma I_{n_u} & 0 \\ -L^T & 0 & -\gamma I_{n_y} \end{bmatrix}<0 \tag{4.29}$$

$$\sum_{i=1}^{s}\dot{\rho}_i\frac{\partial X(\rho)}{\partial\rho_i}+X(\rho)(A(\rho)+LC(\rho))+(A(\rho)+LC(\rho))^TX(\rho)-\beta C^T(\rho)C(\rho)<0 \tag{4.30}$$

$$\begin{bmatrix} Y(\rho) & I_n \\ I_n & X(\rho) \end{bmatrix}<0 \tag{4.31}$$

where $\beta=(\gamma-\gamma^{-1})$.

Proof 3 *These sufficient conditions can be derived in a coprime factorisation framework using Theorem 5. The generalised plant has state-space realisations:*

$$G_s=\begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \tag{4.32}$$

where

$$\begin{aligned} &A=A(\rho),\quad B_1=-L,\quad B_2=B(\rho),\quad C_1=\begin{bmatrix} C(\rho) \\ 0_{n_u\times n}\end{bmatrix}, \\ &C_2=C(\rho),\quad D_{11}=\begin{bmatrix} I_{n_y} \\ 0_{n_u\times n_y}\end{bmatrix},\quad D_{12}=\begin{bmatrix} 0_{n_y\times n_u} \\ I_{n_u}\end{bmatrix},\quad D_{21}=I_{n_y}. \end{aligned} \tag{4.33}$$

Substituting into (4.23), we have (omitting the dependence of ρ*)*

$$\begin{bmatrix} \sum_{i=1}^{s}\dot{\rho}_i\frac{\partial X}{\partial\rho_i}+X(A+LC)+(A+LC^T)X-\gamma C^TC & \begin{bmatrix} C^T & 0\end{bmatrix} \\ \begin{bmatrix} C \\ 0 \end{bmatrix} & \begin{matrix} -\gamma I & 0 \\ 0 & -\gamma I \end{matrix} \end{bmatrix}<0$$

and using the Schur complement, the above inequalities can be further simplified to

$$(A+LC)^TX + X(A+LC) + \sum_{i=1}^{s} \dot{\rho}_i \frac{\partial X}{\partial \rho_i} - \gamma C^TC + \gamma^{-1}C^TC < 0,$$

which is equivalent to (4.30). Similarly, a straightforward substitution allow one to derive (4.29).

Suppose we have X,Y satisfying the necessary and sufficient conditions of Theorem 6, and L is derived from Theorem 4. Let $Q(\rho), S(\rho)$ be invertible matrix functions that satisfy

$$X(\rho)Y(\rho) + Q(\rho)S^T(\rho) = I. \tag{4.34}$$

We have controller state space matrices

$$\begin{aligned} A_K(\rho) &= -Q(\rho)^{-1}[A^T(\rho) + X(\rho)A(\rho)Y(\rho) - \gamma X(\rho)B(\rho)B^T(\rho) \\ &\quad -\gamma C^T(\rho)C(\rho)Y + X(\rho)LC(\rho)Y(\rho) + \gamma^{-1}C^T(\rho)C(\rho)Y(\rho) \\ &\quad +C^T(\rho)L^T + X(\rho)\dot{Y}(\rho) + \dot{Q}(\rho)S(\rho)]S^{-T} \qquad (4.35) \\ B_K(\rho) &= Q^{-1}(\rho)(X(\rho)L - \gamma C^T(\rho)) \\ C_K(\rho) &= -\gamma B^T(\rho)S(\rho)^{-T} \end{aligned}$$

which define a strictly proper ($D_k = 0$) dynamic output feedback controller that solves the γ-performance loop shaping design problem in terms of a coprime factorisation of the LPV plant. This control formula is derived from substituting equations (4.33) into the control formula in (4.12).

Because of the infinite number of constraints in (4.29)–(4.31), we again need to approximate the LPV plant by griding the parameter space to relieve the infinite-dimensionality of the constraints. The parameter-dependent matrices X and Y can be parameterised by a combination of continuously differentiable functions that describes the system dynamics. The advantage of using Wu's PDLF design framework is that the controller formulas are explicitly given. This makes the direct online implementation of the controller possible and it allows us to treat the gain-scheduling controllers as a single entity, with the gain scheduling achieved entirely by the parameter-dependent controller.

The design procedure for the loop shaping/PDLF method can be summarised as

- Follow the McFarlane-Glover loop shaping design procedure; shape the nominal plant with pre and post-compensators to get $G_S(\rho)$.

- Grid the parameter space, select the appropriate parameterisation for $X(\rho)$ and $Y(\rho)$, and choose the bound on the rate of variation $\mathcal{V}_\rho$.

- Use Theorem 4 and solve the LMIs at each grid point to obtain L, and derive the coprime factorisation interconnection structure.

- Compute the Lyapunov matrices $X(\rho), Y(\rho)$ and the scalar variable γ as solutions to the LMI systems (4.29)–(4.30).

- If the LMIs are feasible, use the formulas in (4.35) to construct the controller.

- The final controller can be constructed using the output feedback controller and compensators with the formula $W_1 K(\rho) W_2$.

4.3 Incorporating Pole Placement Constraints

The synthesis of a controller using the PDLF method with a performance index can tackle the control objectives of disturbance attenuation, robust stabilisation and shaping of the open-loop response. The method is similar to H_∞ design and deals mostly with frequency domain aspects. It would therefore be useful if it were possible to incorporate pole placement constraints into the PDLF design to provide control over transient behaviour. According to Chilali and Gahinet [24], satisfactory performance can often be achieved by forcing the poles into so called $\mathcal{D}$-stable regions. They show that the conditions for placing the poles in the convex region can be expressed in terms of LMIs; and mixed state feedback H_2/H_∞ synthesis with pole placement constraints can be solved by satisfying additional LMIs. In [48], the pole placement constraints have been used in an LPV controller synthesis for a 6DoF vehicle, and in [36], the pole placement constraints were used in a reduced order parameter dependent LPV synthesis for an F-16 aircraft.

A practical problem in the design of LPV controllers is the presence of fast dynamics, which appear as fast poles for each frozen LTI system in the parameter variation set. This imposes digital implementation restrictions because of the "aliasing" effects. The appearance of these fast dynamics, which is a common problem involves in the controller synthesis using LMI technique, is probably due to numerical problems in the LMI solutions [24]. As suggested in [25, 24], fast controller dynamics can be prevented by adding pole placement constraints into the original LMI synthesis problem. The poles of the closed-loop system can then be forced into a D-stable region. This is an *ad hoc* approach that assumes "frozen" LTI pole placement will benefit the time response of the LPV system and eliminate undesirable pole locations.

The synthesis method we develop is based on the results in [24] for LTI H_∞ synthesis with pole clustering. We incorporate the LMI pole clustering conditions into Wu's PDLF framework. The following definition is taken from [24].

Defintion 5 *LMI Region: A subset $\mathcal{D}$ is called an LMI region if there exist a symmetric matrix $\alpha = [\alpha_{kl}] \in \mathbf{R}^{m\times m}$ and a matrix $\beta = [\beta_{kl}] \in R^{m\times m}$ such that*

$\mathcal{D} = \{z \in \mathbf{C} : f_{\mathcal{D}} < 0\}$ *with*

$$f_{\mathcal{D}}(z) = \alpha + z\beta + \bar{z}\beta^T = [\alpha_{kl} + \beta_{kl}z + \beta_{lk}\bar{z}\beta^T]_{1\leq k,l\leq m} \tag{4.36}$$

This region makes up a dense subset in the set of regions of the complex plane, symmetric with respect to the real axis. It is therefore appealing for specifying pole placement design objectives. The pole location in a given LMI region can be characterised in terms of the $m \times m$ block matrix

$$M_{\mathcal{D}}(A, W) := [\alpha_{kl}W + \beta_{kl}W + \beta_{lk}WA^T]_{1\leq k,l\leq m} \tag{4.37}$$

Theorem 7 *The matrix A is $\mathcal{D}$ stable if and only if there exist a symmetric matrix X such that*

$$M_{\mathcal{D}}(A, W) < 0, \quad X > 0.$$

where W is given in the definition 4.

The convex region we use in our design is a vertical strip defined by:

$$\mathcal{D} = \{x + jy \in \mathbf{C} : \quad -h_1 < x < -h_2 < 0\}. \tag{4.38}$$

Corollary 1 *Given two LMI regions $\mathcal{D}_1$ and $\mathcal{D}_2$, a matrix A is both $\mathcal{D}_1$ stable and $\mathcal{D}_2$ stable if and only if there exists a positive definite matrix W such that $M_{\mathcal{D}_1}(A, W) < 0$ and $M_{\mathcal{D}_2}(A, W) < 0$.*

A proof of the theorem 7 and Corollary 1 can be found in [24]. Suppose we have the LPV system defined in (3.13) and closed-loop system matrices $A_{clp}(\rho)$ as in (3.14). The pole placement constraints can be characterised by the following lemma.

Lemma 12 *The LPV system in (3.13) has quadratic performance of level γ and closed-loop poles lie in the region of (4.38) if some parameter dependent matrix function $W(\rho) = W(\rho)^T$ satisfies the following matrix inequalities:*

$$\begin{bmatrix} A_c^T(\rho)W(\rho) + W(\rho)A_c(\rho) + \sum_{i=1}^{s} \dot{\rho}_i \dfrac{\partial W(\rho)}{\partial \rho_i} & W(\rho)B_c(\rho) & C_c^T(\rho) \\ B_c^T(\rho)W(\rho) & -\gamma I_{n_d} & D_c^T(\rho) \\ C_c(\rho) & D_c(\rho) & -\gamma I_{n_e} \end{bmatrix} < 0 \tag{4.39}$$

$$A_c(\rho)W(\rho) + W(\rho)A_c^T(\rho) + 2h_2W(\rho) < 0, \tag{4.40}$$

$$A_c(\rho)W(\rho) + W(\rho)A_c^T(\rho) + 2h_1W(\rho) > 0. \tag{4.41}$$

This result is a direct application of Corollary 1. More polynomial expressions for different convex regions can be found in [49]. To characterise pole placement constraints in Wu's PDLF framework, we have the following theorem which is able to place the closed-loop pole in the vertical strip region defined by (4.38).

Theorem 8 *The γ-performance output feedback problem with pole placement constraints characterized in (4.38) can be solved by adding the following LMIs to the original LMIs (4.2)–(4.4) for the output feedback synthesis:*

$$2h_2 \begin{bmatrix} Y(\rho) & I \\ I & X(\rho) \end{bmatrix} + (S(\rho) + S^T(\rho)) < 0 \tag{4.42}$$

$$2h_1 \begin{bmatrix} Y(\rho) & I \\ I & X(\rho) \end{bmatrix} + (S(\rho) + S^T(\rho)) > 0 \tag{4.43}$$

with the short hand notation

$$S = \begin{bmatrix} YA^T - \gamma B_2B_2^T - Y^TC_{12}^TB_2^T & B_2C_{12} + B_{12}C_2 - \gamma^{-1}(YC_{11}^TC_{11} + B_{11}B_{11}^TX) \\ A^T & A^TX - C_2^TB_{12}^TX^T - \gamma C_2^TC_2 \end{bmatrix}. \tag{4.44}$$

For the loop shaping/PDLF method, this expression can be further simplified to

$$S = \begin{bmatrix} YA^T - \gamma BB^T & -LC - \gamma^{-1}YCC^T \\ A^T & A^TX + C^TL^TX^T - \gamma C^TC. \end{bmatrix} \tag{4.45}$$

Proof 4 *Omitting the dependence on ρ, we define $XY + MN^T = I$ and*

$$\begin{aligned} W &= \begin{bmatrix} X & N \\ N^T & Q^{-1} \end{bmatrix} \quad W^{-1} = \begin{bmatrix} Y & M \\ M^T & * \end{bmatrix} \\ \Pi_1 &= \begin{bmatrix} Y & I \\ M^T & 0 \end{bmatrix} \quad \Pi_2 = \begin{bmatrix} I & X \\ 0 & N^T \end{bmatrix}. \end{aligned} \tag{4.46}$$

From $WW^{-1} = I$, it can be readily verified that $W\Pi_1 = \Pi_2$. Here we have that M and N are full row rank. If we pre- and post-multiply Eq.(4.40) with Π_1^T and Π_1 respectively, this yields

$$\Pi_1^T A_c W \Pi_1 + \Pi_1^T W A_c^T \Pi_1 + 2h_1 \Pi_1^T W \Pi_1 < 0,$$

which is equivalent to

$$\Pi_1^T A_c \Pi_2 + \Pi_2^T A_c^T \Pi_1 + 2h_1 \Pi_1^T \Pi_2 < 0.$$

Now use A_c in (3.15), Π_1, Π_2 in (4.46) and the controller formula in [23], we get

$$\Pi_1^T A_c \Pi_2 = \left[\begin{array}{cc} T_1 & T_2 \\ T_3 & T_4 \end{array} \right]$$

where

$$\begin{aligned} T_1 &= Y\hat{A}^T - \gamma B_2 B_2^T, \\ T_2 &= (-A + B_2 D_{12}^T C_1 + B_1 D_{21}^T C_2) + \gamma^{-1}(Y C_{12}^T D_{12}^T C_1 + B_1 D_{21}^T B_{12} X^T) \\ &\quad -\gamma^{-1}(Y C_1^T C_1 + B_1 B_1^T), \\ T_3 &= A^T, \\ T_4 &= \tilde{A}^T X - \gamma C_2^T C_2, \end{aligned}$$

with the notations given in Theorem 3. Some algebraic calculations lead to

$$\Pi_1^T A_c \Pi_2 + \Pi_2^T A_c^T \Pi_1 = (S(\rho) + S^T(\rho))$$

and

$$2h_1 \Pi_1^T \Pi_2 = 2h_1 \left[\begin{array}{cc} Y(\rho) & I \\ I & X(\rho) \end{array} \right].$$

Similarly, we can use the same procedure to verify (4.43).

Note that the expression for S in (4.44) is not an LMI because it is not linear in X, Y and γ. A way to overcome this problem is to use an estimated value of γ in the LMI [36]. We can run the synthesis without pole placement constraints to obtain a value for γ, and then use this value in the synthesis with the pole placement constraints.

Chapter 5

LPV Modelling of the UAV Demonstrator

5.1 Generalised Nonlinear Model of the UAV

Modelling is an important process prior to successful aircraft controller design. In this chapter, we present the full nonlinear model of the flying demonstrator UAV "Eclipse" as developed by Cranfield University [55]. The longitudinal and lateral-directional subsystems will be discussed. Then we transform the nonlinear model into an LPV model. This is a prerequisite for the LPV control of nonlinear systems. The approach we employ in the design will be Jacobian linearisation and we validate the LPV model by comparing the responses of the original nonlinear model with the LPV model. Finally, we will analyse the stability and dynamics of the LPV model.

5.1.1 6DoF Nonlinear Model of the UAV

The Eclipse is characterised by being un-manned and tailless with a cropped diamond wing as shown in Figure 5.1. It is powered by a single jet engine and fitted with a retractable undercarriage. The aerodynamic control surfaces consist of wing trailing edge flaps and a rudder on the vertical fin. There is no horizontal stabilizer. Pitching control is achieve by symmetrical deflection of inboard trailing edge flaps.

Differential outboard trailing edge flap deflection provides row control and rudder deflection is in place for yawing control [55].

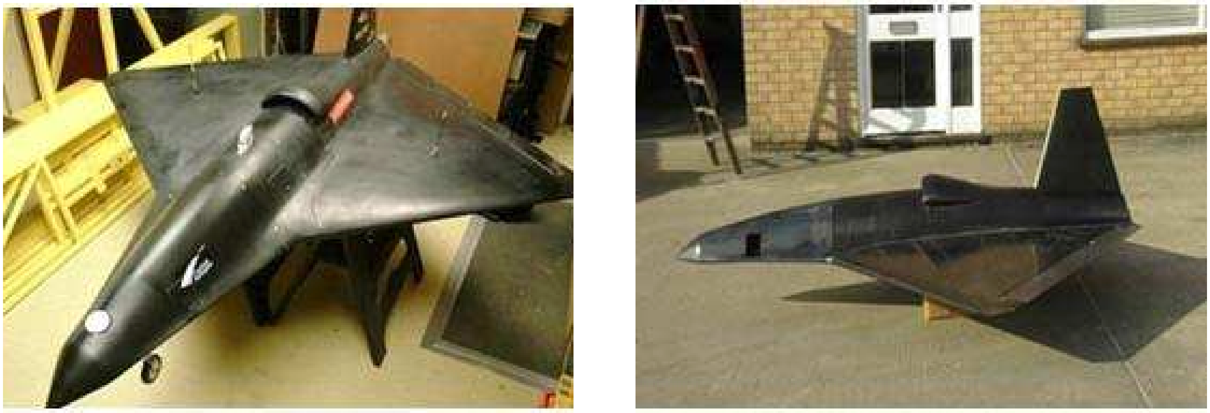


Figure 5.1: “Eclipse” UAV Demonstrator

The aircraft is a rigid body with constant mass and inertia, the origin of the body axes is coincident with the vehicle’s CG. Here we present the nonlinear six degree-of-freedom equations of motion with respect to the body-fixed reference frame which are given in [55, 56, 57], a transformation from body axes to stability axes and body axes to earth axes can be found in [57]:

Force Equations:

$$\begin{aligned} \dot{u} &= \frac{F_x}{m} - qw + rv \\ \dot{v} &= \frac{F_y}{m} - pw - ru \\ \dot{w} &= \frac{F_z}{m} - pv + qu \end{aligned} \tag{5.1}$$

Moment Equations:

$$\dot{p} = \left(1 - \frac{I_{xz}^2}{I_x I_z}\right)^{-1} \left[pq\left(\frac{I_{xz}}{I_x} + \frac{I_{xz}(I_x - I_y)}{I_x I_z}\right) + qr\left(\frac{I_y - I_z}{I_x} - \frac{I_{xz}^2}{I_x I_z}\right) + \frac{M_x}{I_x} + \frac{I_{xz}}{I_x I_z} M_z\right]$$

$$\dot{q} = pr\frac{I_z I_x}{I_y} - (p^2 - r^2)\frac{I_{xz}}{I_y} + \frac{M_y}{I_y}$$

$$\dot{r} = \left(1 - \frac{I_{xz}^2}{I_x I_z}\right)^{-1} \left[pq\left(\frac{I_x - I_y}{I_z} + \frac{I_{xz}^2}{I_x I_z}\right) + qr\left(-\frac{I_{xz}}{I_z} + \frac{I_{xz}(I_y - I_z)}{I_x I_z}\right) + \frac{I_{xz}}{I_x I_z} M_x + \frac{M_z}{I_z}\right] \tag{5.2}$$

Gravitational terms:

$$\begin{aligned} \dot{u} &= rv - qw - g\text{sin}\theta + \frac{X}{m} \\ \dot{v} &= pw - ru + g\text{sin}\phi\ \text{cos}\theta + \frac{Y}{m} \\ \dot{w} &= qu - pv + g\text{cos}\phi\ \text{cos}\theta + \frac{Z}{m} \end{aligned} \tag{5.3}$$

Kinematic equations:

$$\begin{aligned} \dot{\phi} &= p + q\ \text{sin}\phi\ \text{tan}\theta + r\ \text{cos}\phi\ \text{tan}\theta \\ \dot{\theta} &= q\ \text{cos}\phi - r\ \text{sin}\phi \\ \dot{\psi} &= q\ \text{sin}\phi\ \text{sec}\theta + r\ \text{cos}\phi\ \text{sec}\theta \end{aligned} \tag{5.4}$$

Navigation equations:

$$\begin{aligned} \dot{P}_N &= u\ \text{cos}\theta\ \text{cos}\psi + v(\text{sin}\phi\ \text{sin}\theta\ \text{cos}\psi - \text{cos}\phi\ \text{sin}\psi) \\ &\quad + w(\text{cos}\phi\ \text{sin}\theta\ \text{cos}\psi +\ \text{sin}\phi\ \text{sin}\psi) \\ \dot{P}_E &= u\ \text{cos}\theta\ \text{sin}\psi + v(\text{sin}\phi\ \text{sin}\theta\ \text{cos}\psi - \text{cos}\phi\ \text{sin}\psi) \\ &\quad + w(\text{cos}\phi\ \text{sin}\theta\ \text{sin}\psi +\ \text{sin}\phi\ \text{cos}\psi) \\ \dot{h} &= u\ \text{sin}\theta - v\ \text{sin}\phi\ \text{cos}\theta - w\ \text{cos}\phi\ \cos\theta \end{aligned} \tag{5.5}$$

Note that the force equations are expressed in terms of velocity components in the aircraft body-fixed system. Since the aerodynamic force and moment components in the above equations depend on the aircraft angles and true airspeeds, the replacement transformations can be given by velocity, incidence angle and sideslip angle as

$$
\begin{aligned}
V_t &= \sqrt{u^2 + v^2 + w^2} \\
\beta &= \sin^{-1}\left(\frac{v}{V_t}\right) \\
\alpha &= \tan^{-1}\left(\frac{w}{u}\right)
\end{aligned}
\tag{5.6}
$$

where the range for these three parameters are given according to the wind tunnel test results in [55] as: 22m/s $\leq V_t \leq$ 72m/s, $-20^\circ \leq \beta \leq 20^\circ$ and $1.5^\circ \leq \alpha \leq 15.3^\circ$.

5.1.2 Airframe Dynamics

The airframe is usually divided into two subsystems: axial/longitudinal and lateral/directional. This division is based on the decoupling assumptions between the two linear perturbed subsystems about their trim equilibrium.

Longitudinal system

Aircraft spend most of their flying time in a wing-level steady state flight condition. In this case, the roll angle ϕ is zero. If the sideslip angle β is negligible and the roll and yaw rates are small, we can obtain decoupled equations for pure longitudinal motion [58]. The longitudinal system can be represented by the following linear dynamic equations:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{\theta} \\ \dot{q} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} X_u & X_w & -g\text{cos}\theta_0 & X_q & 0 \\ Z_u & Z_w & -g\sin\theta_0 & Z_q & 0 \\ 0 & 0 & 0 & 1 & 0 \\ M_u & M_w & M_\rho & M_q & 0 \\ \sin\theta_0 & -\cos\theta_0 & 0 & u_0\cos\theta_0 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ \theta \\ q \\ h \end{bmatrix} + \begin{bmatrix} X_\eta & X_\tau \\ Z_\eta & Z_\tau \\ 0 & 0 \\ M_\eta & M_\tau \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \tau \end{bmatrix} \tag{5.7}$$

where the approximate expressions for the aerodynamic stability derivative can be given as in Table 5.1[57].

and the output channels are

$$\begin{bmatrix} q \\ h \\ \gamma_f \\ V_t \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -1/V_t & 1 & 0 & 0 \\ \text{cos}\alpha & \text{sin}\alpha & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ \theta \\ q \\ h \end{bmatrix} \tag{5.8}$$

From the navigation equation, we also have altitude h given by

$$h = \int V_t(\theta - \frac{w}{V_t})dt$$

The perturbed linear model states can be defined as

$$u = \mathbf{u} - u_0, \quad w = \mathbf{w} - w_0, \quad q = \mathbf{q}, \quad , \quad h = \mathbf{h} - h_0, \quad \theta = \mathbf{\Theta} - \theta_0,$$

where u_0, w_0, θ_0, h_0 are the body-axis nominal equilibrium values of airspeed, vertical heave, pitch angle and air vertical height, respectively. The bold value variables $(\mathbf{u}, \mathbf{w}, \mathbf{q}, \mathbf{\Theta}, \mathbf{h})$ are true states of the longitudinal system, γ_f is the flight path angle. η and τ are the elevator deflection and the throttle input respectively.

The main longitudinal aerodynamic coefficients can be characterised as follow:

- Drag coefficient

$$C_D = C_{D_0} + KC_L^2$$

where $C_{D_0} = 0.0172$ is the zero lift drag coefficient, $K = 0.155$ is a general constant.

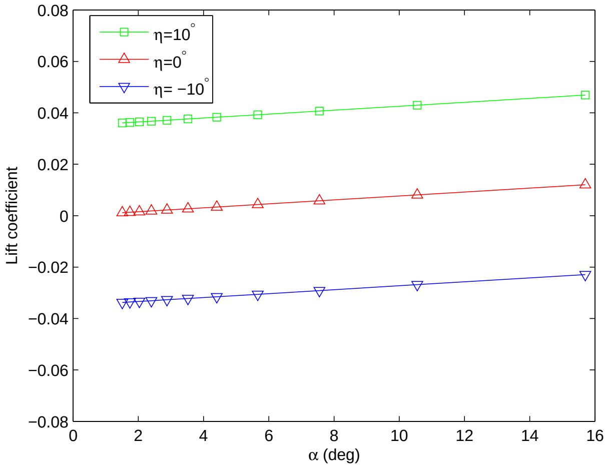


Figure 5.2: Lift coefficient C_L

- Lift coefficient

  $$C_L = C_{L_\alpha}(\alpha - \alpha_0) + C_{L_\eta}\eta$$

  where $C_{L_\alpha} = 2.51$ is the lift curve slop, and $C_{L_\eta} = 0.2$ is the lift coefficient due to elevator deflection.

- Pitching moment coefficient

  $$C_M = C_{L_\alpha}(\alpha - \alpha_0)(X_{cg} - X_{ac_w})\cos\alpha/c_{bar} + C_D(X_{cg} - X_{ac_w})\sin\alpha/c_{bar} + \\ C_{L_\eta}\eta(X_{cp_e} - X_{cg})\cos\alpha/c_{bar} + C_{Mf_\alpha}\alpha$$

  where $X_{cg} = 1.113$ is the position of the center of gravity, $X_{ac_w} = 1.211$ is the aerodynamic center, $X_{cpe} = 1.565$ is the elevator center of pressure, $X_{cpf} = 1.693$ is the flap center of pressure.

A list of aerodynamic data can be found in appendix. We can see C_L, C_D and C_M depend on both α and elevator angle η. A plot of aerodynamics coefficient with respect to both α and η are shown in Figure 5.2–5.4. We can see those aerodynamics coefficient does not vary much with changing angle of attack.

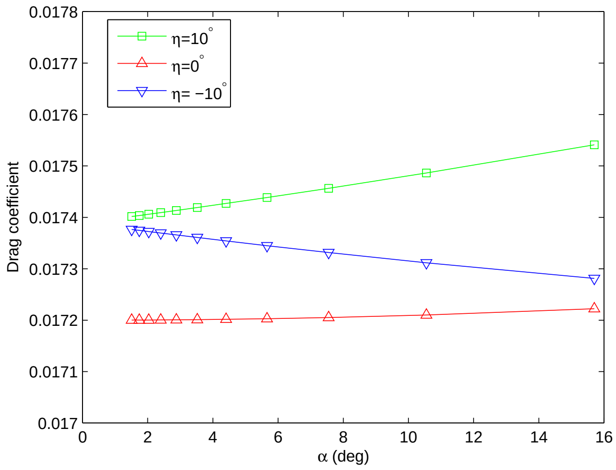


Figure 5.3: Drag coefficient C_D

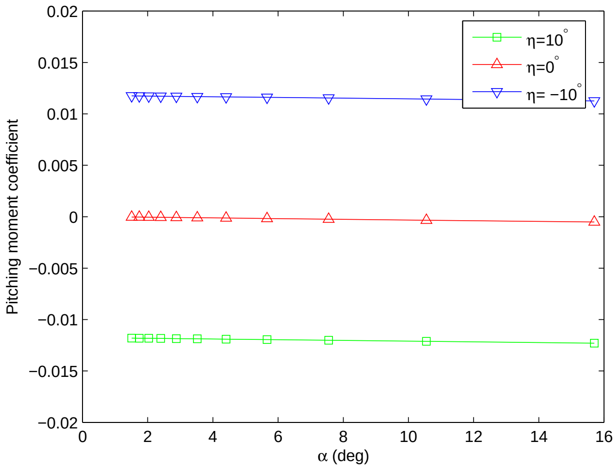


Figure 5.4: Pitching moment coefficient C_M

Lateral-directional System

The lateral/directional system (v, p, r, ϕ, ψ) represents the perturbed states of sideslip, roll rate, yaw rate, roll angle, heading angle. The linear dynamic system is represented by the following equations:

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p + w_0 & Y_r - u_0 & g\cos\Theta_0 & 0 \\ L_v & L_p & L_r & 0 & 0 \\ N_v & N_p & N_r & 0 & 0 \\ 0 & \sec\Theta_0 & \tan\Theta_0 & 0 & 0 \\ 0 & 0 & \sec\Theta_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} Y_\xi & Y_\zeta \\ L_\xi & L_\zeta \\ N_\xi & N_\zeta \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix} \tag{5.9}$$

The perturbed linear model states are defined as:

$$v = \mathbf{v} - v_0, \;\; p = \mathbf{p}, \;\; r = \mathbf{r}, \quad \phi = \mathbf{\Phi} - \phi_0, \quad \psi = \mathbf{\Psi} - \psi_0$$

where the bold variables $(\mathbf{v},\;\; \mathbf{p},\;\; \mathbf{r},\;\; \mathbf{\Phi},\;\; \mathbf{\Psi})$ are unperturbed true respective states. (ξ, ζ) are the aileron and rudder control inputs respectively. The output channels can be characterised as

$$\begin{bmatrix} p \\ r \\ \phi \\ \beta \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/V_t & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} \tag{5.10}$$

where β is sideslip angle.

The lateral aerodynamics are calculated by summing the coefficients contributes to a given force or moment coefficients as follows

- Sideforce coefficient

  $$C_Y = C_{Y_\beta}\frac{v}{V_t} + C_{Y_\zeta}\zeta + C_{Y_r} r\frac{b}{2V_t} + C_{Y_p} p\frac{b}{2V_t}$$

  where $C_{Y\beta}, C_{Y_\xi}, C_{Y_r}$ and C_{Y_p} are side force coefficient due to lateral velocity, aileron deflection, yaw rate and aileron deflection respectively.

- Roll coefficient

$$C_{LL} = C_{LL_\beta}\frac{v}{V_t} + C_{LL_\xi}\xi + C_{LL_\zeta}\zeta + C_{LL_p}p\frac{b}{2V_t} + C_{LL_r}r\frac{b}{2V_t}$$

  where $C_{LL\beta}, C_{LL_\xi}, C_{LL_\zeta}, C_{LL_p}$ and C_{LL_r} are roll coefficient due to due to lateral velocity, aileron deflection, rudder deflection, roll rate and yaw rate respectively.

- Yaw coefficient

$$C_N = C_{N_\beta}\frac{v}{V_t} + C_{N_\zeta}\zeta + C_{N_\xi}\xi + C_{N_p}\frac{b}{2V_t} + C_{N_r}r\frac{b}{2V_t}$$

  where $C_{N\beta}, C_{N_\xi}, C_{N_\zeta}, C_{N_p}$ and C_{N_r} are roll coefficient due to due to lateral velocity, aileron deflection, rudder deflection, roll rate and yaw rate respectively. a list of aerodynamics data for lateral dynamics can be found in appendix.

5.2 Development of the LPV Model

A prerequisite to applying LPV controller synthesis is to transform the nonlinear model of the system into an LPV model. This modelling has become a key issue in the design of LPV controllers. The LPV model can be considered as a group of local descriptions of nonlinear dynamics. Therefore it is essential that the LPV model captures the transient behaviour of the real plant. There are several approaches to obtaining reliable LPV models. Jacobian linearisation [59] is the most widely used in practice and its theoretical development is the most straightforward. Generally, control designers use a family of linear, time-invariant (LTI) plants at different points of interest throughout the operational envelope in order to obtain an LPV model. It is based on first-order approximations with respect to a set of equilibrium points. State transformation is another approach. This is based on differentiable functions of the non-scheduling states and control inputs with the goal of removing any nonlinear terms not dependent on scheduling parameters [60]. Another LPV modelling approach is function substitution, which uses a substitution of a so-called decomposition function by a linear combination of functions and scheduling vectors. A comprehensive survey of these three approaches can be found in [60]. In this section, we will use the Jacobian linearisation approach to develop an LPV model of the UAV.

5.2.1 Numerical Linearisations

Before deriving the trim model, we need to select the scheduling parameters. Generally, the scheduling parameters for an aircraft could be altitude h, velocity V_t, angle of attack α and dynamic pressure $\bar{q}$. Because the flight altitude of the UAV is usually between 1000m and 5000m, the air density effects on the flight dynamics can be neglected. The UAV does not operate at high angles of attack, and so the effects of the incidence angle α on the flight dynamics are limited. Therefore for an efficient controller design using PDLF, we choose the total velocity V_t as the only scheduling parameter.

Next it is necessary to find the equilibrium points of several flight conditions. The flight envelope of interest in this design covers aircraft speeds between 22m/s and 72m/s. We selected 11 different airspeeds spaced every 5m/s in the range to define the design grid. The flight altitude is chosen to be 1000m. The definition of steady wings-level flight assumes that all the motion variables, axial, normal and side forces and the roll, pitch and yaw moments are constant or zero. Hence, provided the aircraft is stable it will remain in equilibrium until it is disturbed by control inputs or external influences such as turbulence. Therefore, the state variables of the nonlinear 6DoF equation can be used to define the steady state conditions as follows:

$$\dot{p}, \dot{q}, \dot{r}, \dot{u}, \dot{v}, \dot{w}, \dot{\beta}, \dot{\alpha} = 0$$

The steady state condition $\dot{p}, \dot{q}, \dot{r} = 0$ requires the angular rates to be zero or constant and therefore aerodynamics and thrust moments to be zero or constant. The condition $\dot{u}, \dot{v}, \dot{w} = 0$ requires the airspeed, incidence and side slip angle to be constant.

For example, the equilibrium solution at $v = 37$m/s for a wings-level flight is

$$[u \;\; v \;\; w \;\; p \;\; q \;\; r \;\; \phi \;\; \theta \;\; \psi \;\; h] = [36.8 \;\; 0 \;\; 3.65 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0.1 \;\; 0 \;\; 1000],$$

and the control inputs are

$$[\eta \;\; \xi \;\; \zeta \;\; \tau]_{eq} = [-9.7414 \;\; 0 \;\; 0 \;\; 0.2224],$$

where η is elevator deflection with a given range $(-30^\circ \leq \eta \leq 30^\circ)$ and τ is the throttle input $(0 \leq \tau \leq 1)$, the minimum throttle demand to avoid engine cut-off is

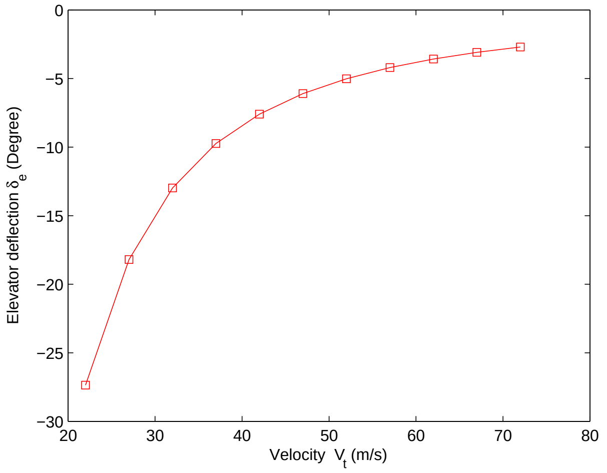


Figure 5.5: Trim characteristics of elevator in terms of velocity

0.08. The equilibrium trim condition has $\alpha_e = 5.67°$. The rudder and aileron inputs are zero for wing-level flight. Trim characteristics of the of elevator and throttle in terms total velocity can be seen in Figure 5.5 and 5.6.

The trim model of the UAV can be obtained by performing linearisations with respect to the set of trim points for the local linear model. Using the Jacobian linearisation, a trim map in terms of scheduled parameters can be found. These state-space realisations of the longitudinal and lateral models together with the associated trim map compose the Jacobian LPV model.

Most commonly the Jacobian matrices are derived in an analytical form if the linearisation can be written in an analytical, well-defined form, and only interpolation of the trim map is necessary. A drawback of this approach is that it is a tedious and error-prone task to find the partial derivatives for the aerodynamic coefficients. We use numerical differentiation in this particular design. More accurate models can be obtained by increasing the grid density albeit with a greater computational demand. We will verify that the linearised models capture the local nonlinearities of the plant in the next section.

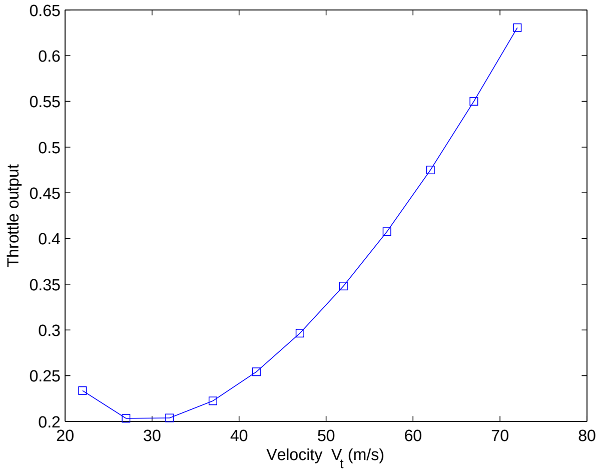


Figure 5.6: Trim characteristics of throttle in terms of velocity

5.2.2 Validation of the Model

In order to assess the accuracy of the linearised process, we will compare the time response of the trim linear model to the full nonlinear model. The simulation is entirely open-loop. For the trim level flight, the lateral-directional states and inputs are mostly constrained to be zero, so we only investigate the longitudinal dynamics here. It is known that an LPV model can be seen as a group of local descriptions of nonlinear dynamics. Therefore, all the command inputs are given as small perturbations from the equilibrium. We consider the trim point $V = 62$m/s where the equilibrium inputs are defined as $[\eta \quad \xi \quad \zeta \quad \tau]_{eq} = [-3.5841 \quad 0 \quad 0 \quad 0.4751]$. The following command inputs are applied to the open-loop nonlinear plant and the linearised model:

1. 0.1 step of throttle input starting from t=10s

$$\tau = \begin{cases} 0.4751 & t < 10s \\ 0.6501 & 10s \leq t < 50s \end{cases}$$

2. ±1° of elevator deflections applied from t=20s

$$\eta = \begin{cases} -3.5841^\circ & t < 20s \\ -4.5841^\circ & 20s \leq t < 35s \\ -2.5841^\circ & 35s \leq t < 50s \end{cases}$$

Figures 5.7 and 5.8 show the time responses of the linearised model and the nonlinear model. It is observed that the responses of the linearised model follow the the nonlinear model with some steady offsets when the system is dominated by the phugoid mode. However, the responses capture the short period oscillations rather well considering the various numerical approximations involved in the linearisation. The rapid oscillations of q and w can be explained that the system is dominated by short period mode and exhibit short period pitching oscillation.

For further validation of the linearised model, we consider another operating point $V_t = 42$m/s. The equilibrium trim inputs are given as $[\eta \quad \xi \quad \zeta \quad \tau]_{eq} = [-7.5940 \;\; 0 \;\; 0 \;\; 0.2542]$. The command inputs applied to the model are similar to the first example, i.e.

$$\tau = \begin{cases} 0.2542 & t < 10s \\ 0.3542 & 10s \leq t < 50s \end{cases}$$

Figure 5.9 shows the comparison of the responses. We can see that at $V_t = 42$m/s, the linearised model follows the nonlinear model more accurately than at $V_t = 67$m/s. All the transient characteristics are well captured. This is probably because $V_t = 42$m/s is somehow in the middle of the trim map. Figure 5.10 shows the combined throttle and elevator inputs.

In conclusion, it can be said that the validation results for the linearised model give confidence that it will be useful for the design of the controller. The LTI model derived using Jacobian linearsation captures the nonlinearity of the original model quite well, considering the numerical approximation in the derivations.

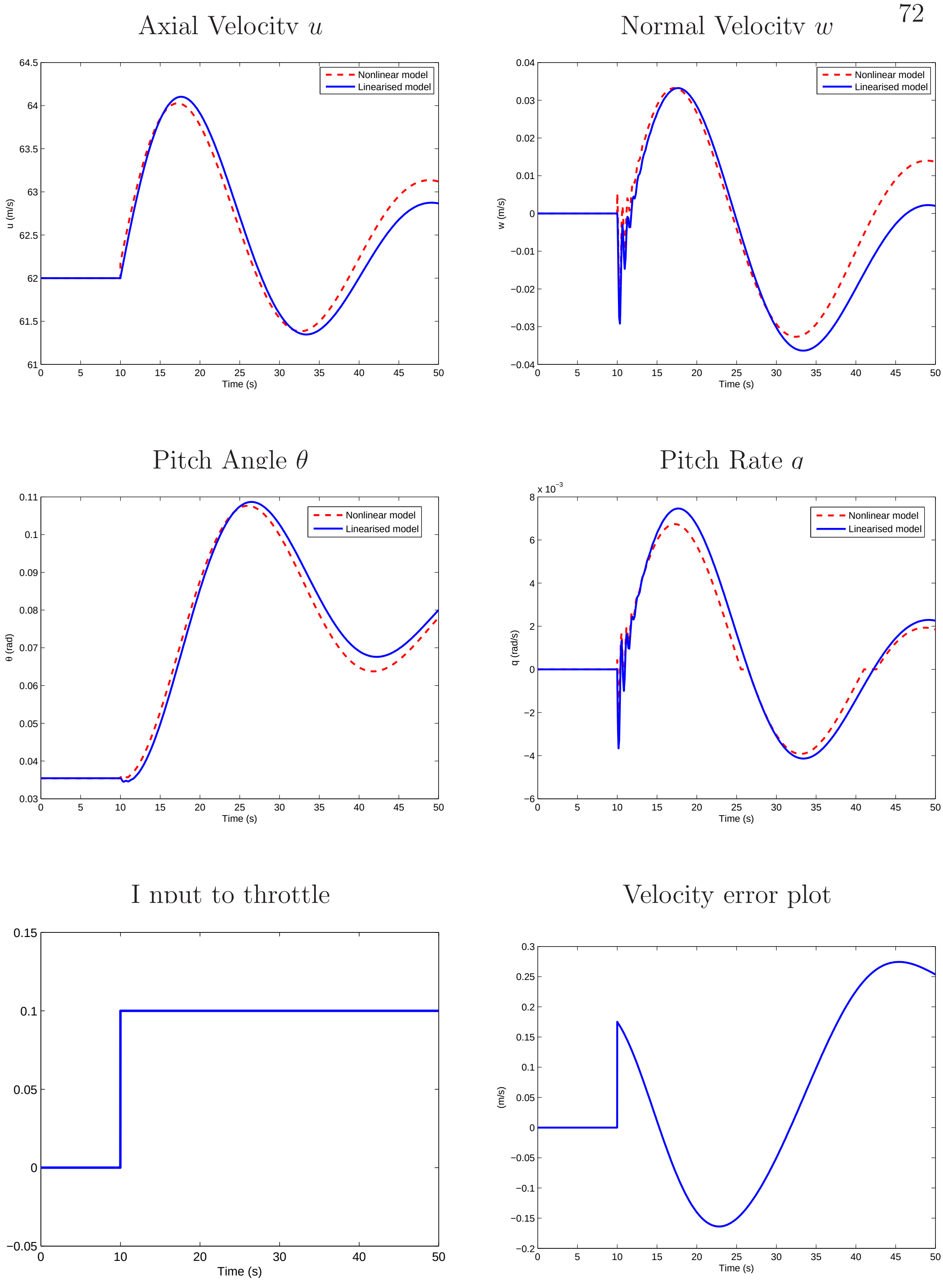


Figure 5.7: Time response for 0.1 step input of τ at $V = 62\text{m/s}$

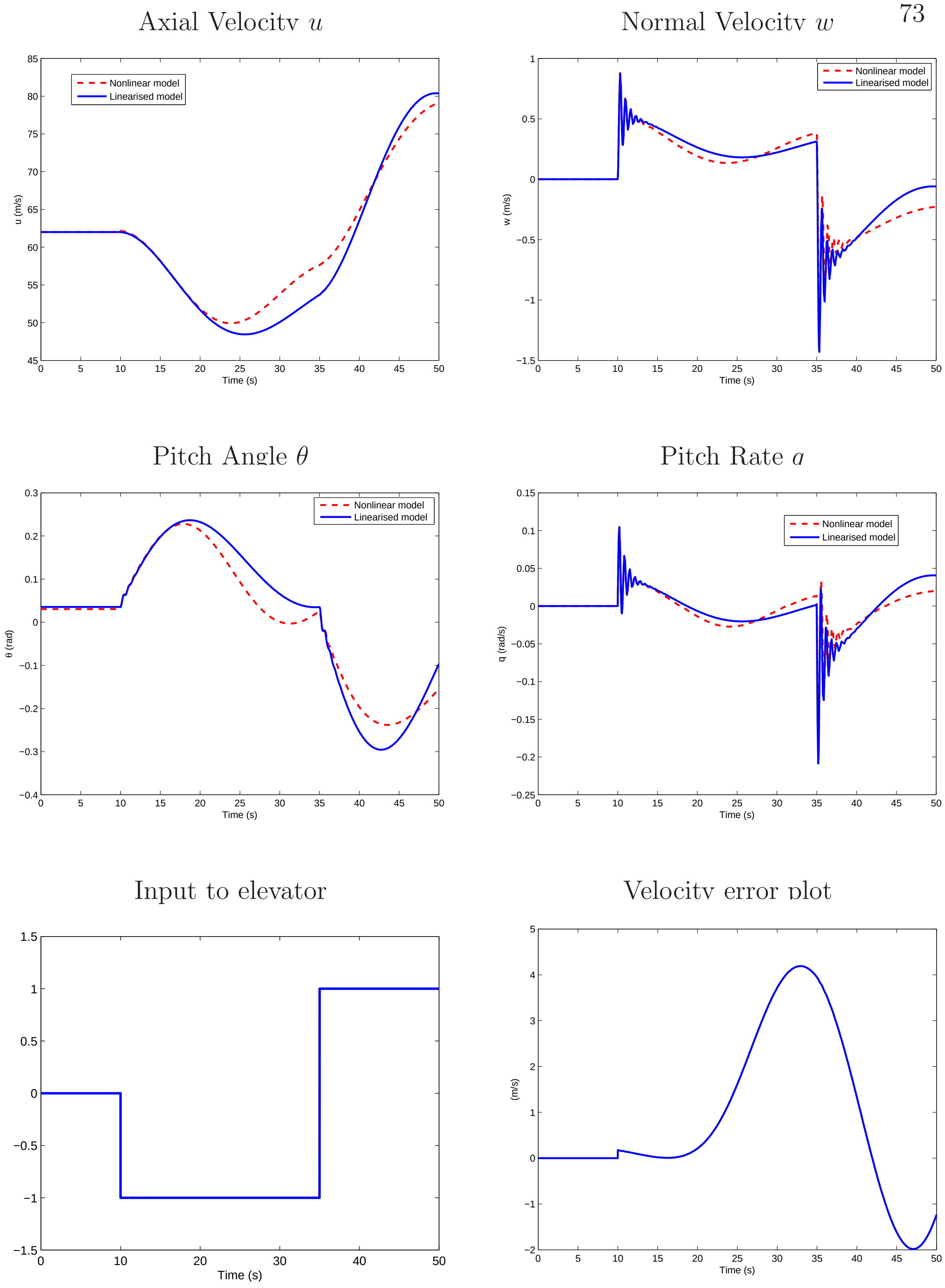


Figure 5.8: Time response for ±1 doublet input of η at $V = 62$m/s

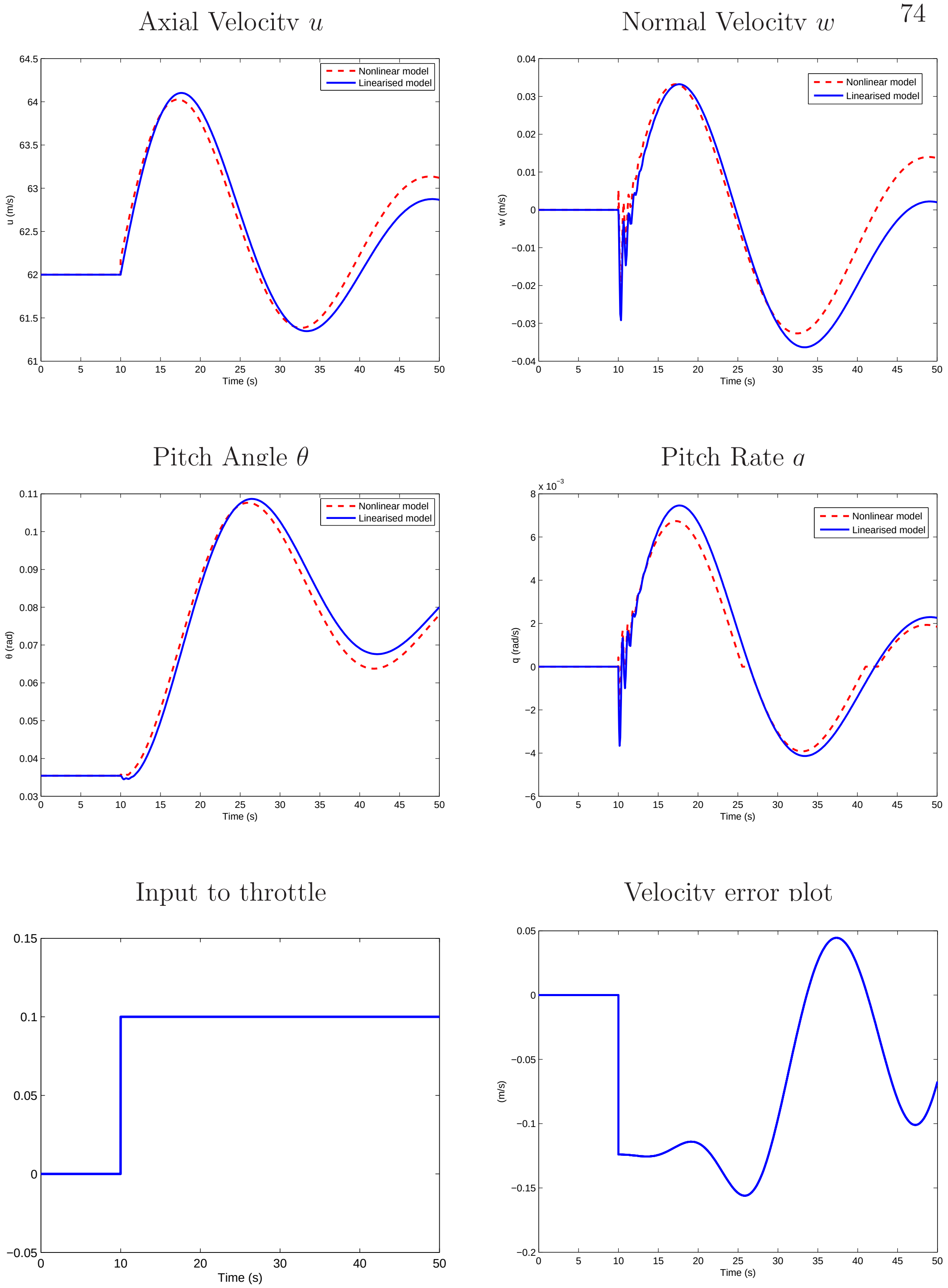


Figure 5.9: Time response for 0.1 step input of τ at $V = 42\text{m/s}$

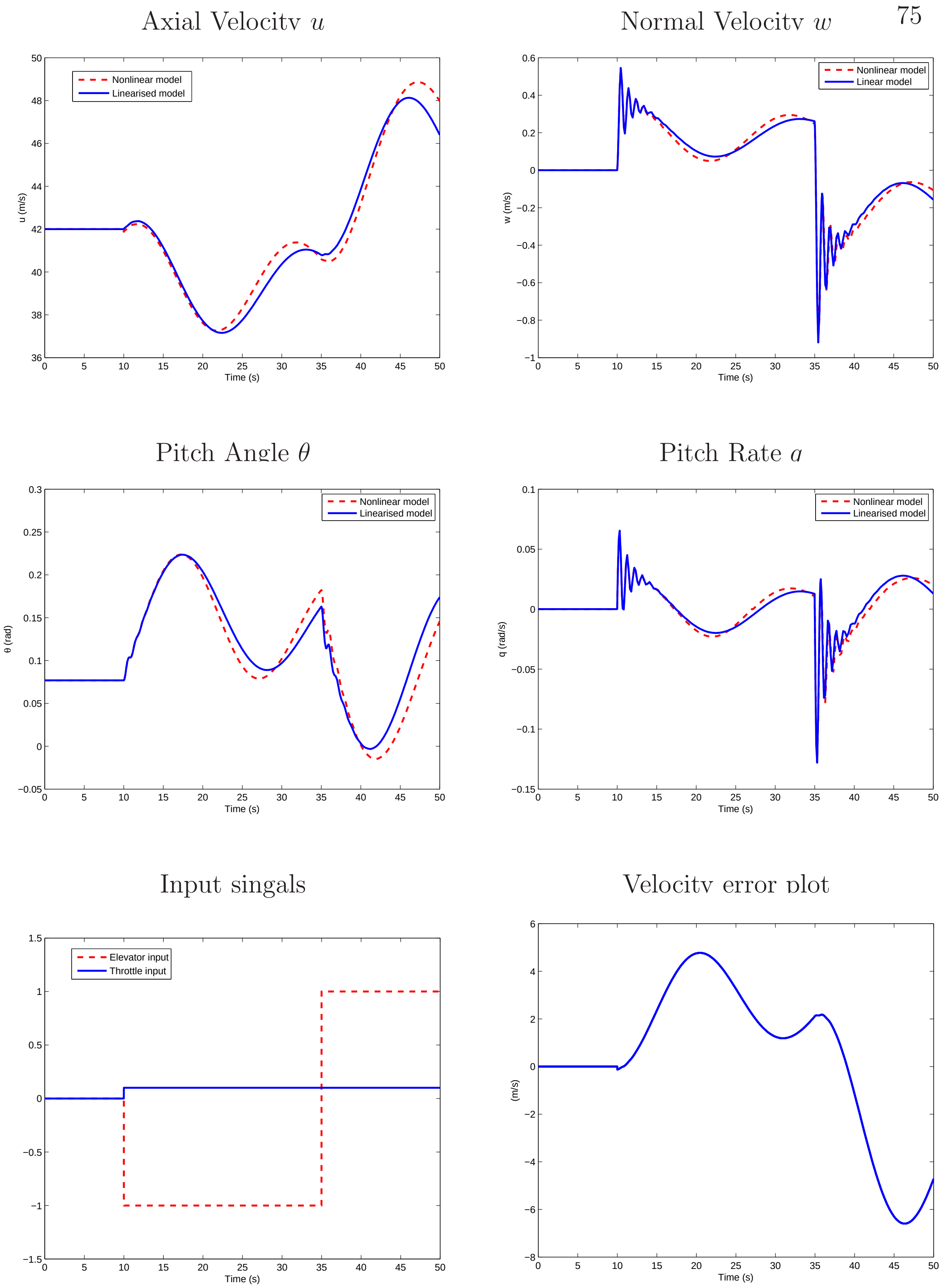


Figure 5.10: Time response for ±1 doublet input of η and 0.1 step input of τ at $V = 67\text{m/s}$

5.3 Dynamic analysis

In this section, we will examine the stability and dynamic characteristics of the longitudinal model of the UAV. If we take the trim point $V_t = 37$ m/s as an example, the longitudinal state-space equation can be given as

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.0445 & 0.3616 & -9.7621 & -3.6516 \\ -0.2209 & -3.1194 & -0.9682 & 36.8194 \\ 0 & 0 & 0 & 1 \\ 0.0493 & -1.0415 & -0.1052 & 1.6383 \end{bmatrix} \begin{bmatrix} u \\ w \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 0.0052 & 4.9765 \\ -0.1584 & 0 \\ 0 & 0 \\ -0.2884 & -0.2969 \end{bmatrix} \begin{bmatrix} \eta \\ \tau \end{bmatrix} \tag{5.11}$$

From the equation given in (5.7), the longitudinal equation (5.11) exhibits the following static stability characteristics:

- stable axial speed disturbance: $X_u < 0$.
- stable vertical speed disturbance: $Z_w < 0$
- asymmetric thrust: $M_u > 0$
- negative longitudinal stability margin: $M_w < 0$

Dynamic stability can be evaluated by analysing the system's characteristic equation given as:

$$s^4 + 1.5256s^3 + 33.6667s^2 + 1.8992s + 3.7437 = 0.$$

The bare airframe poles representing the roots of the characteristic function are shown in Figure 5.11. The pair of roots at 0.73rad/s represent the short period mode, whilst the pair close to the origin depict the phugoid mode. Approximate values of damping ratio and undamped natural frequency are calculated as

Phugoid mode:

Damping ratio	$\zeta_p = 0.0775$
Undamped natural frequency	$\omega_p = 0.334$rad/s

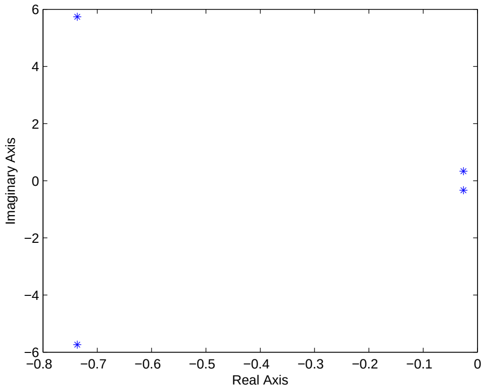


Figure 5.11: Longitudinal open-loop dynamics at 37m/s

where the analytical formula for phugoid frequency and damping can be approximated by [57]

$$\omega_p = g\sqrt{2}/V_t$$
$$\zeta_p = (\sqrt{2}C_L/D_D)$$

Short period mode:

Damping ratio	$\zeta_p = 0.127$
Undamped natural frequency	$\omega_p = 5.79\text{rad/s}$

where the analytical formula for short-period frequency and damping can also by approximated as

$$\omega_p = (-M_\alpha)^{1/2}$$
$$\zeta_p = -(M_{\dot{\alpha}} + M_q)/(2\omega_p)$$

We can see that the phugoid oscillations of the UAV model are poorly damped which can be verified by simulations in the previous section.

The lateral-direction state-space model at $V_t = 37$m/s can be described as follows

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.3711 & 3.6616 & -36.2482 & 937621 \\ -4.9394 & -15.0283 & 9.9646 & 0 \\ 0.7599 & 0.3217 & -4.1760 & 0 \\ 0 & 1 & 0.0992 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & 0.1107 \\ 1.7836 & -1.2555 \\ -0.0089 & 0.4127 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix} \tag{5.12}$$

The static stability characteristics are

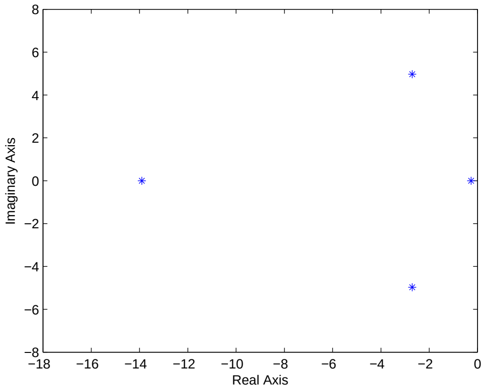


Figure 5.12: Lateral open-loop dynamics at 37m/s

- stable sideslip motion: $Y_v < 0$.
- stable directional motion: $N_v > 0$
- positive stability margin and stable spiral mode: $l_v < 0$

The system's characteristic equation is given as:

$$s^4 + 1.5256s^3 + 33.6667s^2 + 1.8992s + 3.7437 = 0$$

The roots of the equation are shown in Figure 5.12 The real roots that are close to the origin describe the stable spiral mode with time constant

$$T_s = \frac{1}{0.265} = 3.7736\text{s}.$$

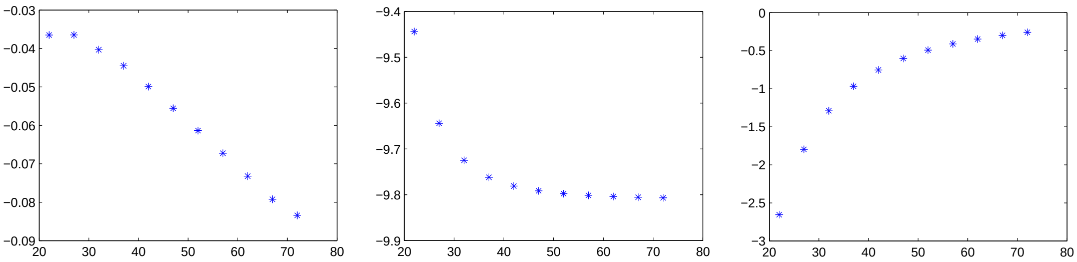
The second real root describes the roll subsidence mode with time constant

$$T_r = \frac{1}{13.9} = 0.0719\text{s}.$$

The pair of complex roots describes the oscillatory dutch roll mode with

Damping ratio	$\zeta_p = 0.478$
Undamped natural frequency	$\omega_p = 5.66\text{rad/s}$

From the discussion in the previous chapter, we know that in order to decide on a parameterisation of the Lyapunov variables, the controller design using the PDLF approach requires some knowledge on how the nonlinear plant depends on the scheduling



A(1,1) A(1,3) A(2,3)

Figure 5.13: Parameter dependence of the system matrix of the UAV

parameters. Figure 5.13 shows the parameter variation of some entries of A matrix. It is observed that the control and stability derivatives vary both linearly and quadratically in the scheduling parameter. We will investigate the parameter dependence of the longitudinal dynamics by constructing an affine parameter dependent model using method of least square curve fit and analysing the dynamics of the affine parameter dependent model. It is also possible to investigate which derivatives are affected most by the varying parameter. This is done by examining the changes in each control and stability derivatives to find out which particular coefficients varying substantially with velocity. It is found that varying velocity introduces significant changes to the coefficients $X_q, Z_\theta, Z_q, M_u, Z_\eta$ and M_η [61]. Considering the characteristics of these coefficients, the variation in X_q and Z_θ mainly affects the damping and frequency of the phugoid mode. The variation in M_u affects primarily the short period frequency and the variation in Z_η and M_η mainly affects the amplitude of the dynamic responses of the aircraft. Thus, by using first order and second order approximation, an affine parameter dependent LPV model of the longitudinal dynamics can be given by

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = A(V_t) \begin{bmatrix} u \\ w \\ \theta \\ q \end{bmatrix} + B(V_t) \begin{bmatrix} \eta \\ \tau \end{bmatrix} \tag{5.13}$$

with matrices $A(V_t)$ and $B(V_t)$ given by

$$A(V_t) = \begin{bmatrix} X_u & X_w & X_\theta & X_q \\ Z_u & Z_w & Z_\theta & Z_q \\ 0 & 0 & 0 & 1 \\ M_u & M_w & M_\theta & M_q \end{bmatrix} \tag{5.14}$$

where each components of matrices are given as

$$\begin{aligned}
X_u &= -0.00102V_t - 0.009115 \\
X_w &= 0.00018V_t^2 + 0.025V_t + 1.0499 \\
X_\theta &= -9.67 \\
X_q &= -0.00166V_t^2 + 0.2308V_t - 10.037 \\
Z_u &= -8.66 \times 10^{-5}V_t^2 + 0.01249V_t - 0.5699 \\
Z_w &= -0.08199V_t - 0.10375 \\
Z_\theta &= -0.001358V_t^2 + 0.1679V_t - 5.4519 \\
Z_q &= 1.012V_t - 0.77 \\
M_u &= 2.6146V_t^2 - 0.003426V_t + 0.1425 \\
M_w &= -0.02817V_t + 0.001135 \\
M_\theta &= -0.0001475V_t^2 + 0.01824V_t - 0.5922 \\
M_q &= 0.04614V_t - 0.08368
\end{aligned}$$

$$B(V_t) = \begin{bmatrix} 0.026 & 24.76 \\ \begin{array}{l} 0.0001159V_t^2 + 4.04 \times 10^5 V_t \\ -0.001415 \end{array} & 0 \\ 0 & 0 \\ \begin{array}{l} 0.00021V_t^2 - 0.00015V_t \\ +0.005065 \end{array} & -0.294 \end{bmatrix} \tag{5.15}$$

The derivation of the affine parameter dependent model involve numerous approximations. We will compare the dynamic characteristics of the linearised model to that of the affine parameter dependent model. The state-space matrices of the affine parameter dependent model and the Linearised model derived from Jacobian linearisation at 22m/s, 47m/s and 72m/s are given below

$$A_{l22} = \begin{bmatrix} -0.036 & 0.614 & -9.443 & -5.953 \\ -0.343 & -1.949 & -2.654 & 21.179 \\ 0 & 0 & 0 & 1 \\ 0.083 & -0.617 & -0.288 & 0.896 \end{bmatrix} \quad B_{l22} = \begin{bmatrix} 0.0053 & 5.096 \\ -0.056 & 0 \\ 0 & 0 \\ -0.099 & -0.304 \end{bmatrix} \tag{5.16}$$

$$A_{a22} = \begin{bmatrix} -0.032 & 0.587 & -9.670 & -5.142 \\ -0.337 & -1.908 & -2.415 & 21.494 \\ 0 & 0 & 0 & 1 \\ 0.080 & -0.619 & -0.262 & 0.931 \end{bmatrix} \quad B_{a22} = \begin{bmatrix} 0.005 & 4.952 \\ -0.057 & 0 \\ 0 & 0 \\ -0.100 & -0.294 \end{bmatrix} \tag{5.17}$$

where (A_{l22}, B_{l22}) and (A_{a22}, B_{a22}) denote the LPV model and the affine parameter dependent model respectively at 22m/s.

$$A_{l47} = \begin{bmatrix} -0.055 & 0.284 & -9.791 & -2.886 \\ -0.174 & -3.943 & -0.602 & 46.911 \\ 0 & 0 & 0 & 1 \\ 0.039 & -1.323 & -0.065 & 2.096 \end{bmatrix} \quad B_{l47} = \begin{bmatrix} 0.005 & 4.927 \\ -0.255 & 0 \\ 0 & 0 \\ -0.466 & -0.293 \end{bmatrix}, \tag{5.18}$$

$$A_{a47} = \begin{bmatrix} -0.057 & 0.272 & -9.791 & -3.276 \\ -0.1740 & -3.9573 & -0.560 & 46.794 \\ 0 & 0 & 0 & 1 \\ 0.039 & -1.323 & -0.061 & 2.096 \end{bmatrix} \quad B_{a47} = \begin{bmatrix} 0.0052 & 4.952 \\ -0.255 & 0 \\ 0 & 0 \\ -0.466 & -0.294 \end{bmatrix}, \tag{5.19}$$

where (A_{l47}, B_{l47}) and (A_{a47}, B_{a47}) denote the system matrices of the LPV model and the affine parameter dependent model respectively at the speed of 42m/s , and finally

$$A_{l72} = \begin{bmatrix} -0.083 & 0.186 & -9.806 & -1.897 \\ -0.113 & -6.025 & -0.258 & 71.97 \\ 0 & 0 & 0 & 1 \\ 0.028 & -2.026 & -0.028 & 3.223 \end{bmatrix} \quad B_{l72} = \begin{bmatrix} 0.0053 & 4.818 \\ -0.598 & 0 \\ 0 & 0 \\ -1.095 & -0.287 \end{bmatrix}, \tag{5.20}$$

$$A_{a72} = \begin{bmatrix} -0.083 & 0.183 & -9.670 & -1.412 \\ -0.119 & -6.007 & -0.403 & 72.094 \\ 0 & 0 & 0 & 1 \\ 0.032 & -2.027 & -0.044 & 3.223 \end{bmatrix} \quad B_{a72} = \begin{bmatrix} 0.003 & 4.952 \\ -0.599 & 0 \\ 0 & 0 \\ -1.095 & -0.294 \end{bmatrix}, \tag{5.21}$$

where (A_{l72}, B_{l72}) and (A_{a72}, B_{a72}) denote the system matrices of the LPV model and the affine parameter dependent model respectively at 72m/s. From the above state-space matrices, it can be seen that all the components of the system matrices remain at more or less the same values with different velocities, except the coefficients $X_q, Z_\theta, Z_q, M_u, Z_\eta$ and M_η. Here we also investigate the undamped natural frequency and damping of the LPV model and the affine parameter dependent model at different velocities. The results are shown in Tables 5.2–5.7.

It is observed that the phugoid undamped natural frequency and damping ratio have very little variation between the linear model and the affine parameter dependent model. Only at the extreme flight speed of 72m/s, is the damping of the phugoid mode of the affine parameter dependent model much less than the damping of the Linear model. In term of the short-period mode, there are some discrepancies between the Linear and affine model. However, at the flight speed of 47m/s, the short period mode characteristics of the affine parameter dependent model are very close to that of Linear model. This is due to the fact that M_a and M_q are always kept at the mean value and it leads to the similar short-period characteristics.

Figures 5.14–5.16 show the comparisons of the frequency responses. It can be observed that at 47m/s and 72m/s, the differences between the frequency responses

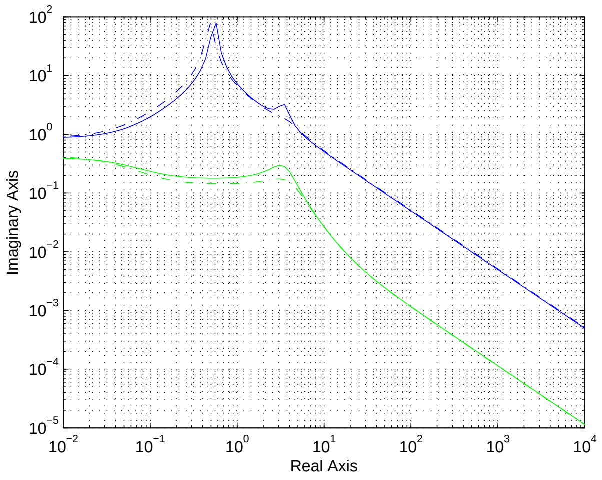


Figure 5.14: Frequency responses of the LPV model and affine model at 22m/s

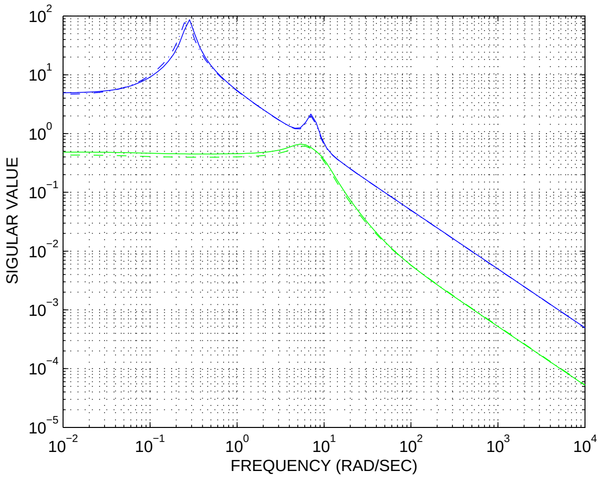


Figure 5.15: Frequency responses of the Affine model and LPV Model at 47m/s

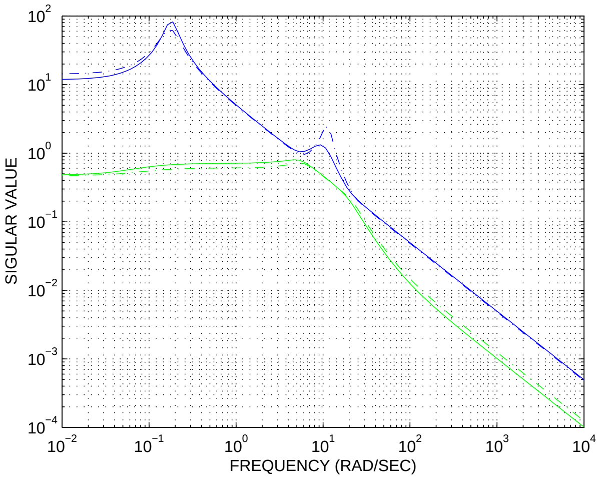


Figure 5.16: Frequency responses of the Affine model and LPV Model at 72m/s

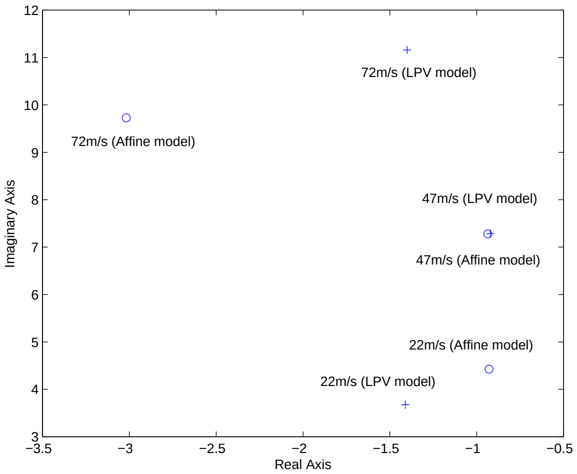


Figure 5.17: Variation of short period eigenvalue with Flight speed for the Affine Model and LPV Model

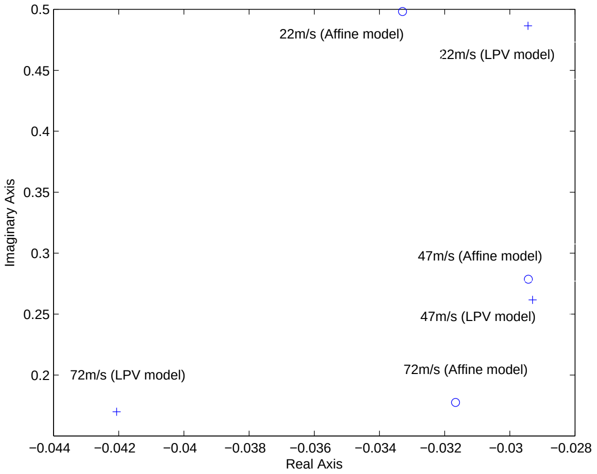


Figure 5.18: Variation of Phugoid eigenvalue with Flight speed for the Affine Model and LPV Model

of the affine parameter dependent model (dotted line) and the Linear model (solid line) are almost negligible. At 22m/s, there are some differences between the pitch rate responses, but the frequency responses of velocity are still very close. Figures 5.17 and 5.18 show the variation of the short-period eigenvalues and the phugoid eigenvalues. It can be seen that only at 47m/s, do both eigenvalues match each other relatively well. This is because most of the derivatives are best approximated around the middle.

Figures 5.19–5.21 represent the open loop responses of the affine parameter dependent models. We observe that at 72m/s and 22m/s, there are some discrepancies between the phugoid oscillations of normal velocity, but at 47m/s, the phugoid oscillation has a good match. This is due to the differences in eigenvalues between two models at 22m/s and 72m/s, but at 47m/s, the difference is negligible .

From the above results, we conclude that the nonlinear dynamics of the UAV depends on the scheduling parameter with linear or quadratic formations. The linear approximations capture most of the dynamic characteristics of the plant except at the extreme flight conditions. Therefore, in the PDLF controller design, we will consider both linear and quadratic parameterisations and do the comparisons.

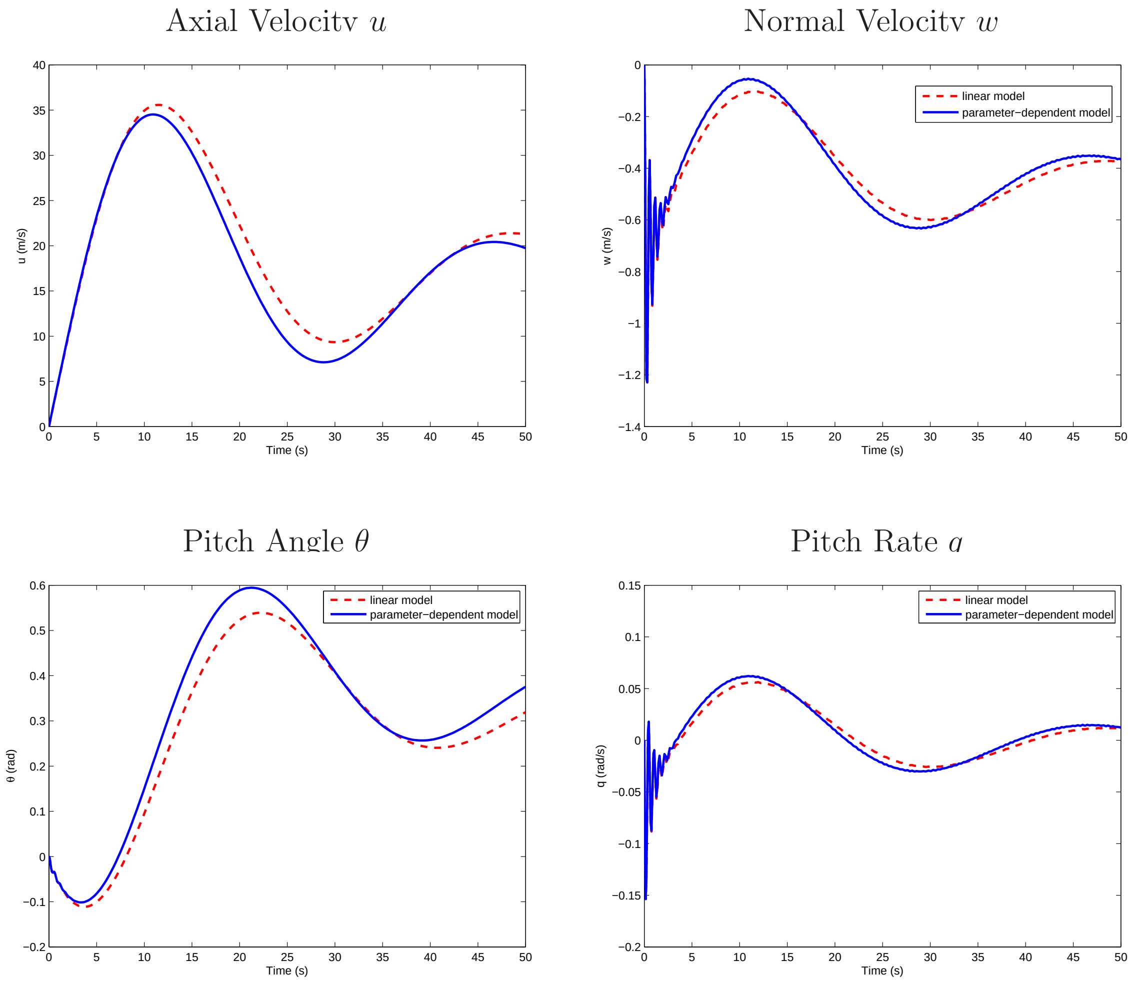


Figure 5.19: Open-loop time responses of the linear model and parameter dependent model at 72m/s

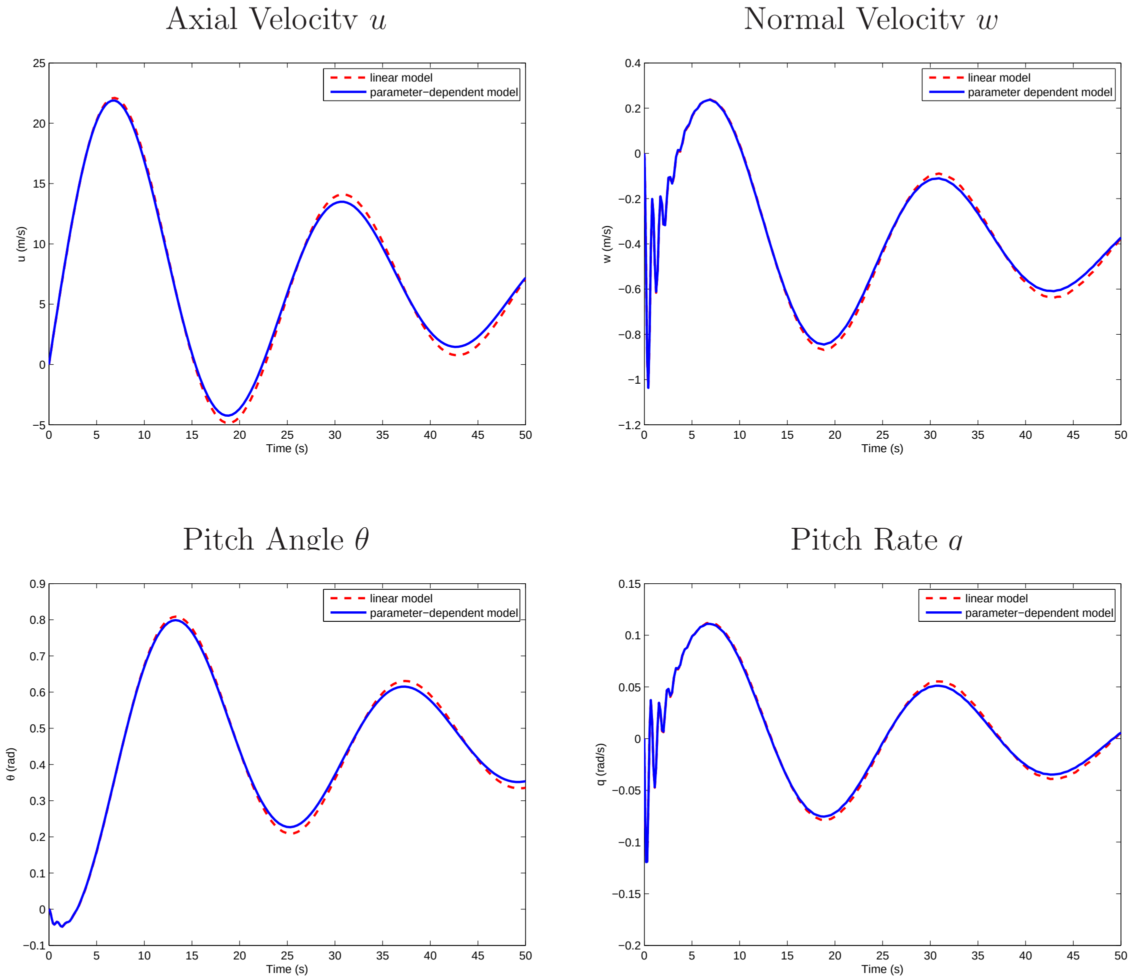
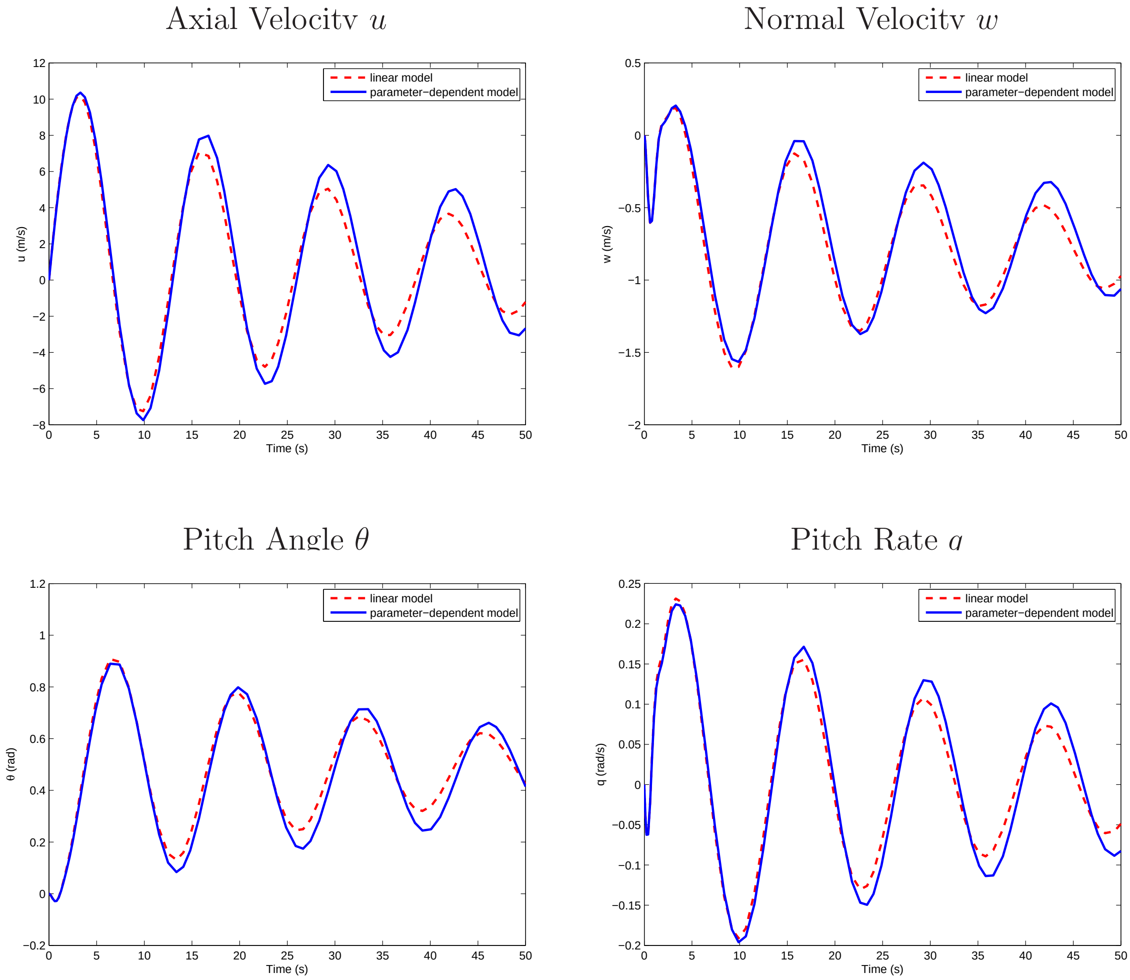


Figure 5.20: Open-loop time responses of the Linear model and parameter dependent model at 47m/s



Axial Velocitv u

Normal Velocitv w

Pitch Angle θ

Pitch Rate q

Figure 5.21: Open-loop time responses of the linear model and parameter-dependent model at 22m/s

Table 5.1: Longitudinal aerodynamic stability derivatives [57]

Derivative	Description	Expression
X_u	Axial force due to velocity	$-2C_D - V_0\dfrac{\partial C_D}{\partial V} + \dfrac{1}{0.5\rho V_0 S}\dfrac{\partial \tau}{\partial V}$
X_w	Axial force due to incidence	$C_L - \dfrac{\partial C_D}{\partial \alpha}$
X_q	Axial force due to pitch rate	$\dfrac{\partial C_D}{\partial \alpha}$
Z_u	Normal force due to velocity	$-2C_L - V_0\dfrac{\partial C_L}{\partial V}$
Z_w	Normal force due to incidence	$-C_D - \dfrac{\partial C_L}{\partial \alpha}$
Z_q	Normal force due to pitch rate	$-V_T a_1$
M_u	Pitching moment due to velocity	$V_0\dfrac{\partial C_m}{\partial V}$
M_w	Pitching moment due to incidence	$-C_{L_\alpha} K_n$
M_q	Pitching moment due to pitch rate	$-V a_1 \dfrac{l_T}{c}$
X_η	Axial force due to elevator	$-2\dfrac{S_T}{S} k_T C_L a_2$
Z_η	Normal force due to elevator	$-\dfrac{S_T}{S} a_2$
M_η	Pitching moment due to elevator	$-V a_2$

Table 5.2: Undamped Natural Frequency at 22m/sec

Model	Phugoid	Short-period
Linear Model	0.566	3.46
Affine Model	0.5	4.52

Table 5.3: Damping at 22m/sec

Model	Phugoid	Short-period
Linear Model	0.058	0.148
Affine Model	0.066	0.206

Table 5.4: Undamped Natural Frequency at 47m/sec

Model	Phugoid	Short-period
Linear Model	0.265	7.35
Affine Model	0.281	7.24

Table 5.5: Damping at 47m/sec

Model	Phugoid	Short-period
Linear Model	0.111	0.125
Affine Model	0.111	0.127

Table 5.6: Undamped Natural Frequency at 72m/sec

Model	Phugoid	Short-period
LPV Model	0.177	11.2
Affine Model	0.197	10.3

Table 5.7: Damping at 72m/sec

Model	Phugoid	Short-period
Linear Model	0.24	0.124
Affine Model	0.16	0.1

Chapter 6

Design Example: A Flying Demonstrator UAV

In this chapter, gain scheduling controllers are designed for a flying demonstrator UAV. The LPV model of the UAV is developed in chapter 5. Firstly the open-loop responses of the model are analysed. We then synthesise a gain-scheduled controller for each of the longitudinal and lateral subsystems using the 2DoF/PDLF method. The following problems will also be investigated:

- Whether the performance of the PDLF method is less conservative than the SQLF method.
- How the parametrisation of the Lyapunov variables affects the level of performance index.
- How the synthesis method with parameter-dependent performance improves the local robust performance when compared to the synthesis that has constant performance level.
- How the pole placement constraints affect the pole locations of the closed-loop system and how the dynamics of the controller are improved.

We demonstrate the linear and nonlinear simulation results of the longitudinal and lateral dynamics. The simulation results can be compared favourably with the performance specifications. The simulation results of a combined flying manoeuvre of longitudinal and lateral motions will also be given.

In the second part of the chapter, we will design an LPV controller using the loop shaping/PDLF synthesis technique. Again we will look at some of the problems that are described above. A test of robustness to model uncertainties will be carried out on the longitudinal fly dynamics. Finally, we show the nonlinear simulation results of both the longitudinal and lateral controllers.

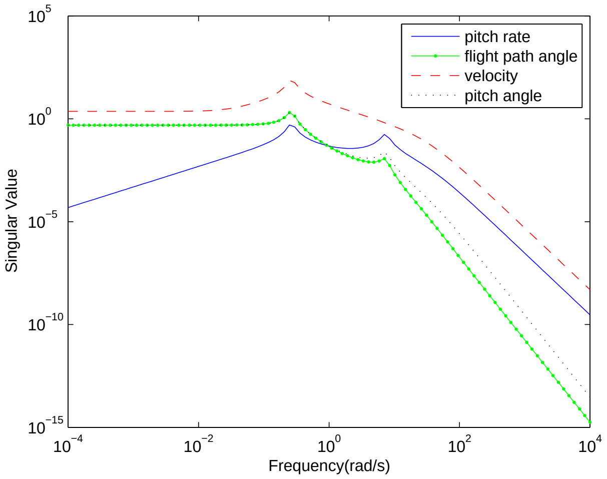


Figure 6.1: Open-loop singular values response from throttle input to each output

6.1 Open-loop plant analysis

In this section, we will study the open-loop singular value in some detail. The LPV model at $V_t = 47$m/s is used in the analysis. The longitudinal and lateral plant models are described in (5.7) and (5.9). Figures 6.1 and 6.2 show the effects of throttle and elevator inputs on all outputs individually in the longitudinal dynamics.

From Figure 6.1, we observe that velocity/throttle loop has low gain across the low frequency band. This is because the engine power was designed to overcome drag and not to have high speed acceleration capability. We also notice that in the high frequency band, the velocity/throttle loop has relatively slow roll-off which should be addressed in the control design to eliminate high frequency activity in the throttle channels.

In Figure 6.2, it is surprising to see that the velocity/elevator loop gain is actually slightly higher than the velocity/throttle loop. This is due to the strong coupling effects between the axial and longitudinal dynamics. We also notice that pitch rate has relatively high gain at high frequency in both figures. This may introduce instability due to the high frequency noise in the measurements.

Figures 6.3 and 6.4 show the effects of the aileron and rudder inputs on all outputs

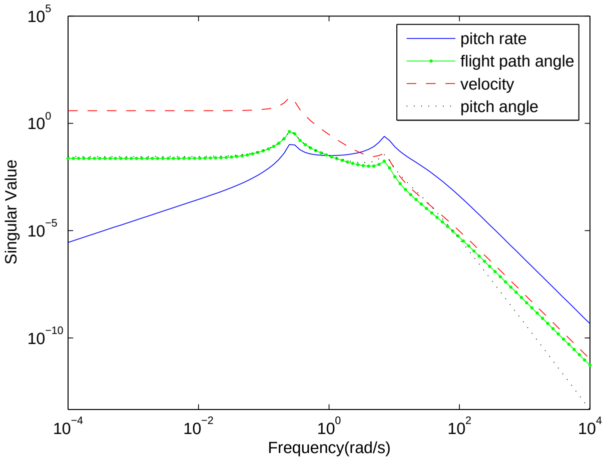


Figure 6.2: Open-loop singular values response from elevator input to each output

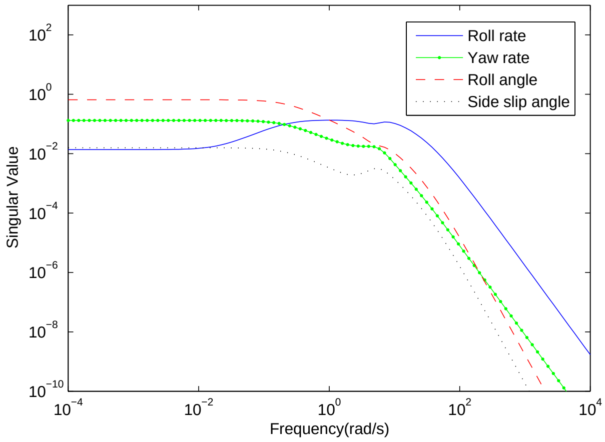


Figure 6.3: Open-loop singular values response from aileron input to each output

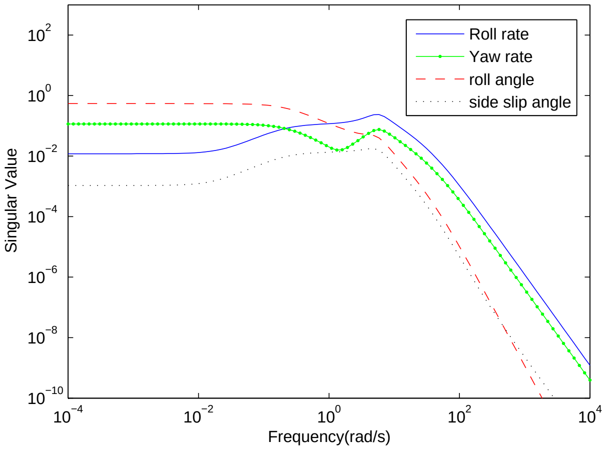


Figure 6.4: Open-loop singular values response from rudder input to each output

individually in the lateral-directional airframe dynamics. In the low frequency band, the aileron and rudder have similar open-loop gains, which justifies the coupling effects between the two control channels. From Figure 6.3, we can see that the aileron is somehow more effective in yaw than roll at frequencies > 0.1rad/sec. The roll rate has relatively high gain at the crossover frequency and might cause the lateral dynamics to become sensitive to uncertainties in the actuator dynamics. In both figures, the low frequency gains of sideslip and roll angle are rather low and have no slope, which indicates nonzero steady state if not dealt with in the design process.

6.2 The Two-degree-of-freedom/PDLF design

6.2.1 The Longitudinal Control Design

In this section, we will develop a 2DoF controller for the UAV demonstrator using the PDLF design method that we discussed in §4.1. The 2DoF closed-loop design structure can be seen in Figure 6.5. Pitch rate (q), flight path angle (γ_f) and total velocity (V_t) are measured and velocity is used for scheduling purposes. The control inputs to the plant are the throttle input and elevator deflection. The reference commands to the system are flight path angle and velocity. It has been shown in the Chapter 4 that the control derivatives do not vary significantly with angle of attack α, and the UAV does not operate at high angle of attack. The flight attitude of the UAV is usually between 1000m and 5000m, therefore we choose velocity as the only scheduled parameter.

Performance objectives and weights

Base on the range and limit specifications in [55], The performance goals of the longitudinal closed-loop system can be characterised as

- Maintain robust stability in the range specified by $22 \leq V_t \leq 72$.
- Track unit step commands of flight path angle with rise time no greater than 1 second, overshoot no greater than 10% and steady state error no greater than 1%.
- Maximum elevator deflection and deflection rate should not exceed 20deg and 35deg/sec respectively.

The plant is augmented by actuators approximated as first order lag filter given as

$$W_{act} = \begin{bmatrix} W_{tho} & 0 \\ 0 & W_{elv} \end{bmatrix} = \begin{bmatrix} \dfrac{5}{s+5} & 0 \\ 0 & \dfrac{20}{s+20} \end{bmatrix},$$

where W_{tho} and W_{elv} denote the throttle and elevator dynamics respectively.

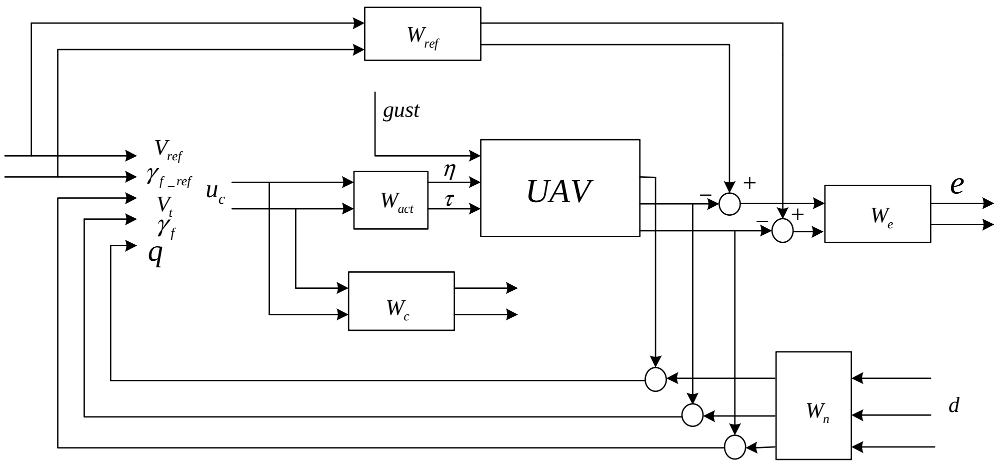


Figure 6.5: The longitudinal flight control design plant

The sensor dynamics are approximated by $\dfrac{50}{s+50}I_3$.

The ideal models that are used in the mixed sensitivity two degree-of-freedom controller design is based on the plant response characteristics. We choose the ideal model to meet the rise time (t_r), settling time (t_s), and overshoot (M_p) for step demand response specifications. The following second order model is chosen for design specifications:

$$M(s) = \frac{\omega_{n2}^2}{s^2 + 2\zeta_2\omega_{n2}s + \omega_{n2}^2}$$

where $\omega_{n2} = 10$rad/s and $\zeta_2 = 10$. In the sequel, the ideal model W_{ref} is given by

$$W_{ref} = \begin{bmatrix} \dfrac{100}{s^2+20s+100} & 0 \\ 0 & \dfrac{100}{s^2+20s+100} \end{bmatrix},$$

where all off-diagonal elements are zero, corresponding to zero cross-coupling.

The performance weight W_e is selected as

$$W_e = \begin{bmatrix} \dfrac{2.5}{s+\epsilon} & 0 \\ 0 & \dfrac{0.714s+71.4}{s+15} \end{bmatrix} \tag{6.1}$$

where $\epsilon = 10^{-3}$. The integral term is chosen for the performance weights of flight path angle. Increasing gain gives the closed-loop system better matching to the ideal model and increased bandwidth of the disturbance rejection.

The disturbance weight W_n is chosen as unit gain matrix. The control weight W_c is used to limit the high frequency activity and to allow low frequency tracking. Hence W_c is selected as a high-pass filter to bound these requirements. A typical choice of the weighting function is $(ks + w)/(s + kw)$, where the parameter w is chosen so that high frequency control is minimised above a certain threshold frequency and k is chosen to allow low frequency control effort. Sensible choices are $k = 10, w = 20$rad/s, and thus

$$W_c = \begin{bmatrix} \dfrac{10s+20}{s+200} & 0 \\ 0 & \dfrac{10s+20}{s+200} \end{bmatrix}. \tag{6.2}$$

We also add constant scaling diag$(1.8, 0.4)$ between reference and error signals for normalising inputs and outputs at about the same magnitude. It is found that by adding the input and output scaling, the γ performance level is effectively reduced.

Computation and synthesis

The augmented closed-loop plant is constructed by combining the linearised model, sensor and actuator models and weighting functions with 18 states: 4 states from the UAV plant model, 2 states from actuators, 3 states from sensors and the remaining 9 states from the weighting functions (4 states from W_{ref}, 2 states from W_c and 2 states from W_e). The system has 10 outputs and 8 inputs. The LPV controller measures 5 outputs (3 from output feedback, and the velocity and flight path angle reference inputs) and generates 2 control inputs (elevator deflections and throttle inputs).

To synthesise a gain-scheduled controller for the lateral subsystem, 11 different velocities spaced every $5m/s$ in the range $22 \leq V_t \leq 72$ are used as the approximated design grid $\mathcal{P}_{grid}$ and an LPV model of the UAV plant is derived according to §5.2. A study of the flight data suggests that a good choice for the set of parameter rates is $\{\dot{V}_t \in \mathcal{V} | -10 \leq \dot{V}_t \leq 10\}$.

With reference to the analysis results of the parameter dependence of the plant dynamics in §5.3, we choose the basis function as

$$g_1(V_t) := 1, \;\; g_2(V_t) := V_t, \;\; g_3(V_t) = V_t^2.$$

A single basis function $f_1(V_t) = 1$ is chosen for X to eliminate the dependence of the controller on $\dot{V}_t$. Therefore the parameter dependence of functions $X(\rho)$ and $Y(\rho)$ are in the form of

$$X(\rho) = f_1(V_t)X_1, \qquad Y(\rho) = \sum_{i=1}^{3} g_i(V_t)Y_i.$$

The synthesis problem is implemented using the Matlab LMI toolbox [63]. We can see a comparison of the PDLF synthesis and SQLF synthesis with constant performance index in Table 6.1.

Table 6.1: Comparison of SQLF and PDLF synthesis

	Number of LMIs	Number of decision variables	LPV γ_{opt}	Computation time
SQLF	23	381	5.808	1 min 9 secs
PDLF	44	761	3.533	3.5 mins

It is observed that the PDLF synthesis has a lower γ performance level than the SQLF synthesis but the computation time is longer. Nevertheless the time difference is not huge due to the fact that we only choose one scheduling parameter. Table 6.2 gives a comparison of different parameterisations of Lyapunov matrix Y in terms of V_t. We can see that the quadratic parameterisation (III) has the lowest value of γ. The linear parameterisation (I) is also a good choice as we predicted in §5.3. The results in Table 6.2 also indicate that the guideline of copying the plant nonlinearities is actually justified.

Table 6.2: Optimal LPV stability margin for different parameterisation

	Parameterisation	LPV optimal γ
I	$Y_0 + V_tY_1$	5.533
II	$Y_0 + V_t^2Y_1$	5.541
III	$Y_0 + V_tY_1 + V_t^2Y_2$	5.533
IV	X, Y constant	5.808

Synthesis with parameter dependent performance

We now present the results of the synthesis, in which the performance measure γ is allowed to be parameter-dependent. The results are shown in the Table 6.3.

Table 6.3: γ at different grid points (SQLF)

V_t	22	27	32	37	42	47	52	57	62	67	72
γ	5.720	5.726	5.729	5.733	5.736	5.738	5.740	5.743	5.769	5.788	5.808

Compared to the constant γ of 6.01 in the worst case measurement, the optimal L_2 gains have been reduced considerably, especially at lower velocity. Therefore better robust performance can be achieved at lower velocity compared to the higher velocity and the overall performance level of the gain-scheduled controller are improved. A similar simulation on the PDLF synthesis yields induced-L_2 performance. The results are shown in Table 6.5. We can see that γ at lower velocity is considerably lower than the constant $\gamma = 3.63$ of the normal PDLF synthesis

In order to compare the performance level, we also synthesise the H infinity controller using *hinfsyn* in the Matlab robust control toolbox for the closed-loop system, the gamma performance level are given as We can see all the performance are around the same level. A plot of performance for three different synthesis methods can be shown in Figure 6.6.

The pole placement approach

In the flight controller design for the UAV, it is found that the PDLF method tends to generate a controller with fast closed-loop poles. We thus add pole placement constraints into the LMI synthesis. The limits of the closed-loop poles are investigated by experiments. Without significantly degrading the γ value, the minimum closed loop pole is moved from -3.2×10^4 to -89 with the convex region $\mathcal{D} = \{x + jy \in \mathbf{C}:$

Table 6.4: Different γ at different grid points (PDLF)

V_t	22	27	32	37	42	47	52	57	62	67	72
γ	5.389	5.407	5.444	5.478	5.483	5.507	5.516	5.522	5.525	5.528	5.530

Table 6.5: Different γ at different grid points (H_∞)

V_t	22	27	32	37	42	47	52	57	62	67	72
γ	5.379	5.373	5.360	5.347	5.335	5.326	5.321	5.316	5.316	5.305	5.300

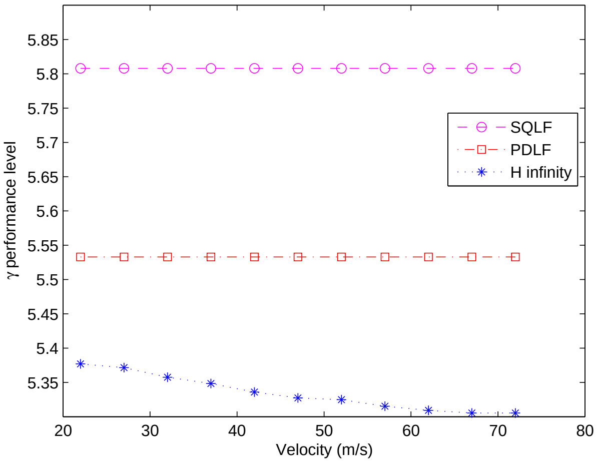


Figure 6.6: Performance under different velocities for PDLF/SQLF/H_∞ methods

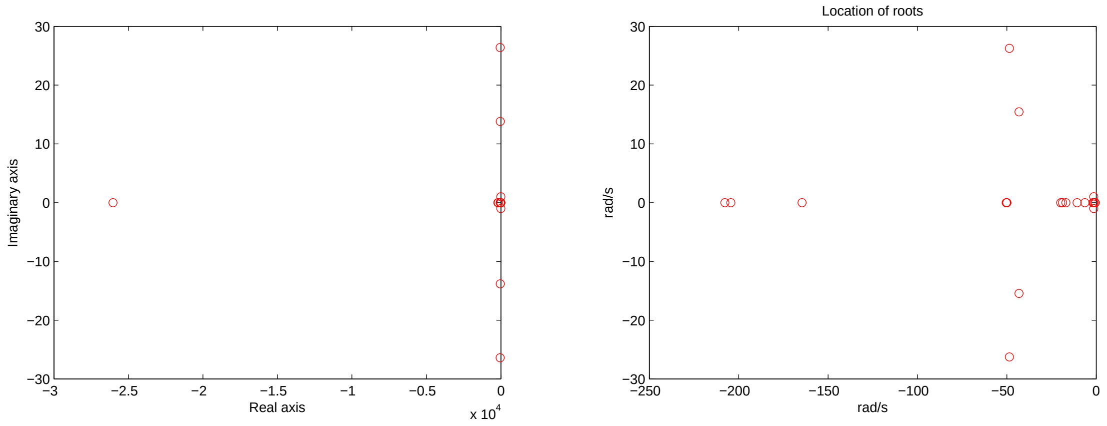
$-100 < x < -0.5 < 0\}$. The results of the robust performance changes are shown in Table 6.6. Further increase in the minimum bound caused the LMIs to become infeasible. The pole locations before and after adding the pole placement constraints can be seen in Figure 6.7.

Table 6.6: Performance level γ vs. minimum pole locations constraints

Pole location constrains	No constraints	-500	-350	-250	-100
LPV γ_{opt}	5.533	5.537	5.538	5.541	5.546

Simulation results

In this section, we apply the controller to the nonlinear model, the actuator saturation limit are set to $-1.5 \le \tau \le 1.5$ and $-15 \le \eta \le 15$. The closed-loop system responses to a unit step demand of velocity and flight path angle at different airspeeds are shown in Figure 6.8. In these simulations, the aircraft is initially



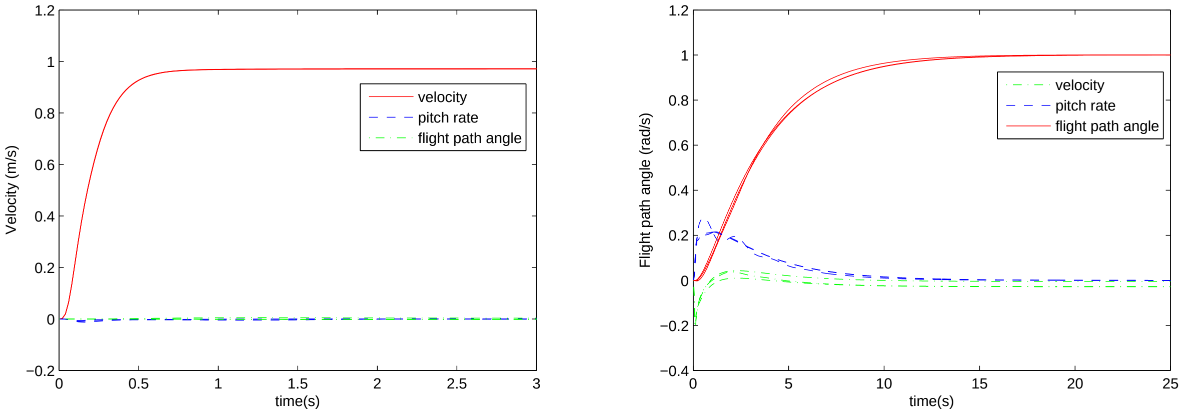
(a) before

(b) after

Figure 6.7: Comparison of the pole locations for 2DoF/PDLF method before and after adding the pole placement constraints

perturbed from trimmed, level flight at the height of 1000m and three different velocities $V_t = 22, 47, 72m/s$. We can see that there are some cross couplings between the responses of fight path angle. For the tracking responses of γ_f, the performance specifications of rise time and overshoot are satisfied.

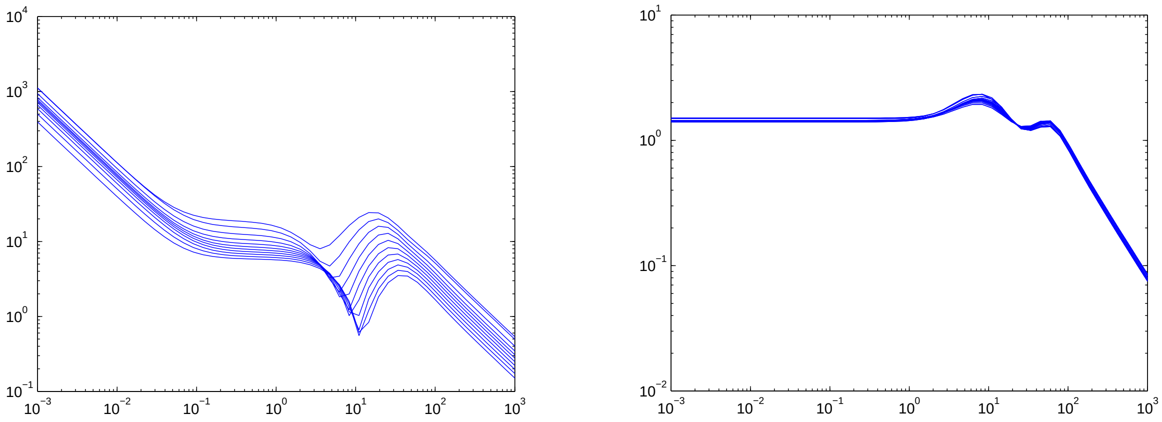
Figure 6.9 shows the frequency responses from the reference demands to the corresponding control surfaces. Figure 6.10 shows the corresponding frequency responses of the closed-loop system. Figures 6.11 and 6.12 demonstrate the nonlinear simulation results of the longitudinal dynamics. Figure 6.11 is a doublet reference tracking response of flight path angle. In order to demonstrate the effect of gain-scheduling, we let velocity track a ramp climbing from 47m/s to 72m/s to cover the whole flight envelope. Figure 6.12 shows a more aggressive tracking of multiple steps of reference of velocity.



(a) Velocity

(b) Flight Path Angle

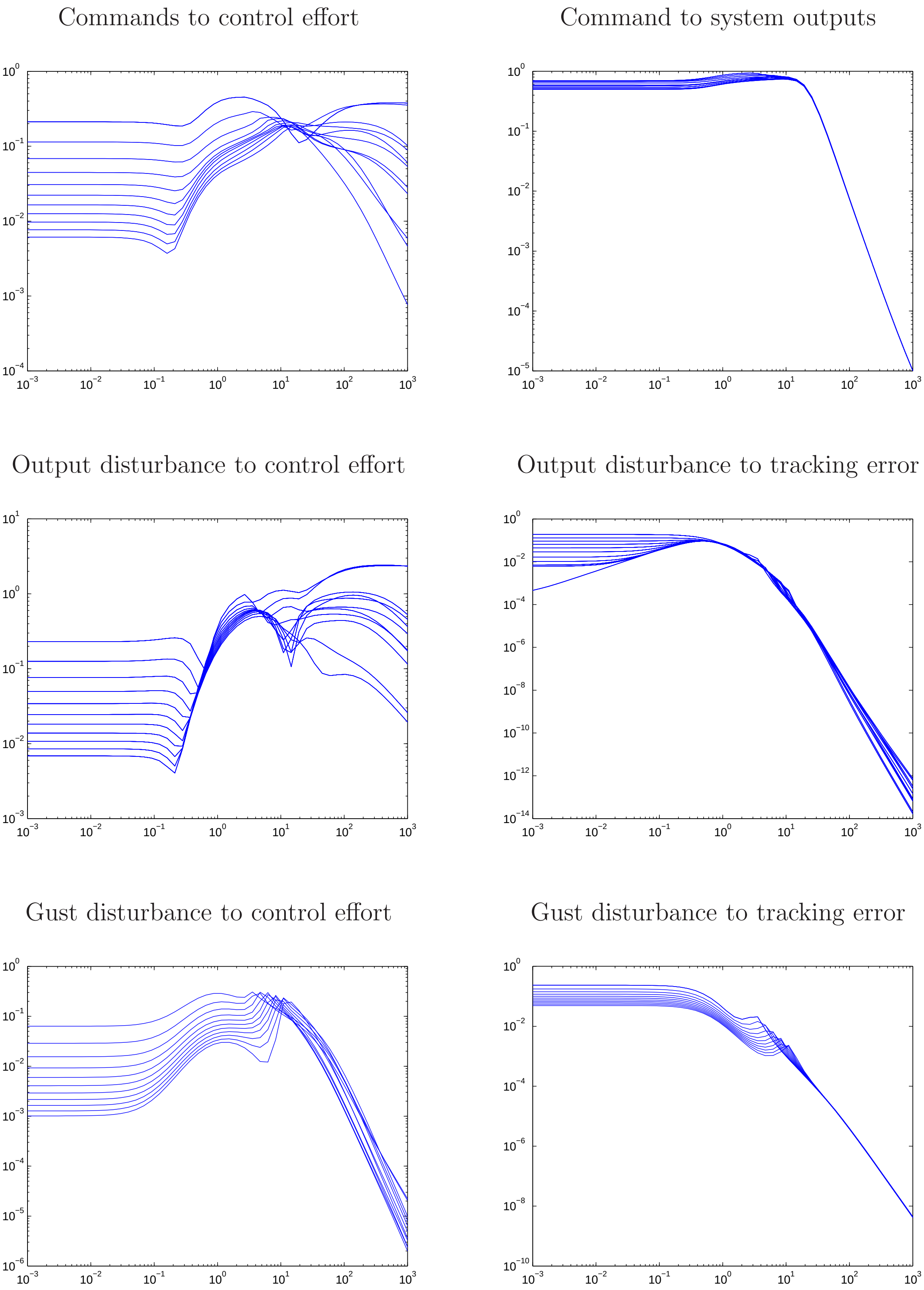
Figure 6.8: Velocity and flight path angle unit step response at 22, 47, 72m/s



Flight path angle reference to elevator

Velocity reference to throttle

Figure 6.9: Frequency response of longitudinal controller



Commands to control effort

Command to system outputs

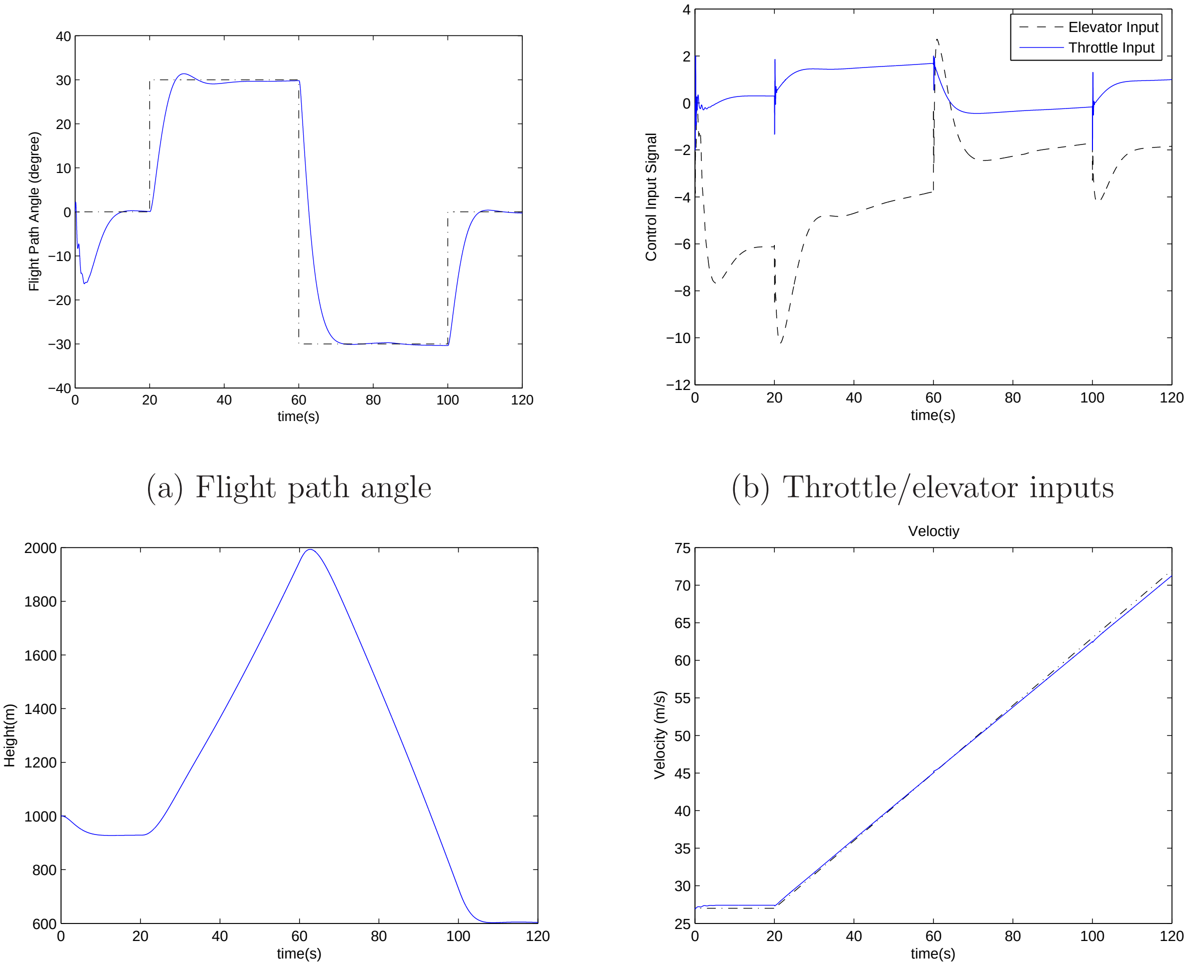
Output disturbance to control effort

Output disturbance to tracking error

Gust disturbance to control effort

Gust disturbance to tracking error

Figure 6.10: Longitudinal closed-loop response by different grid points



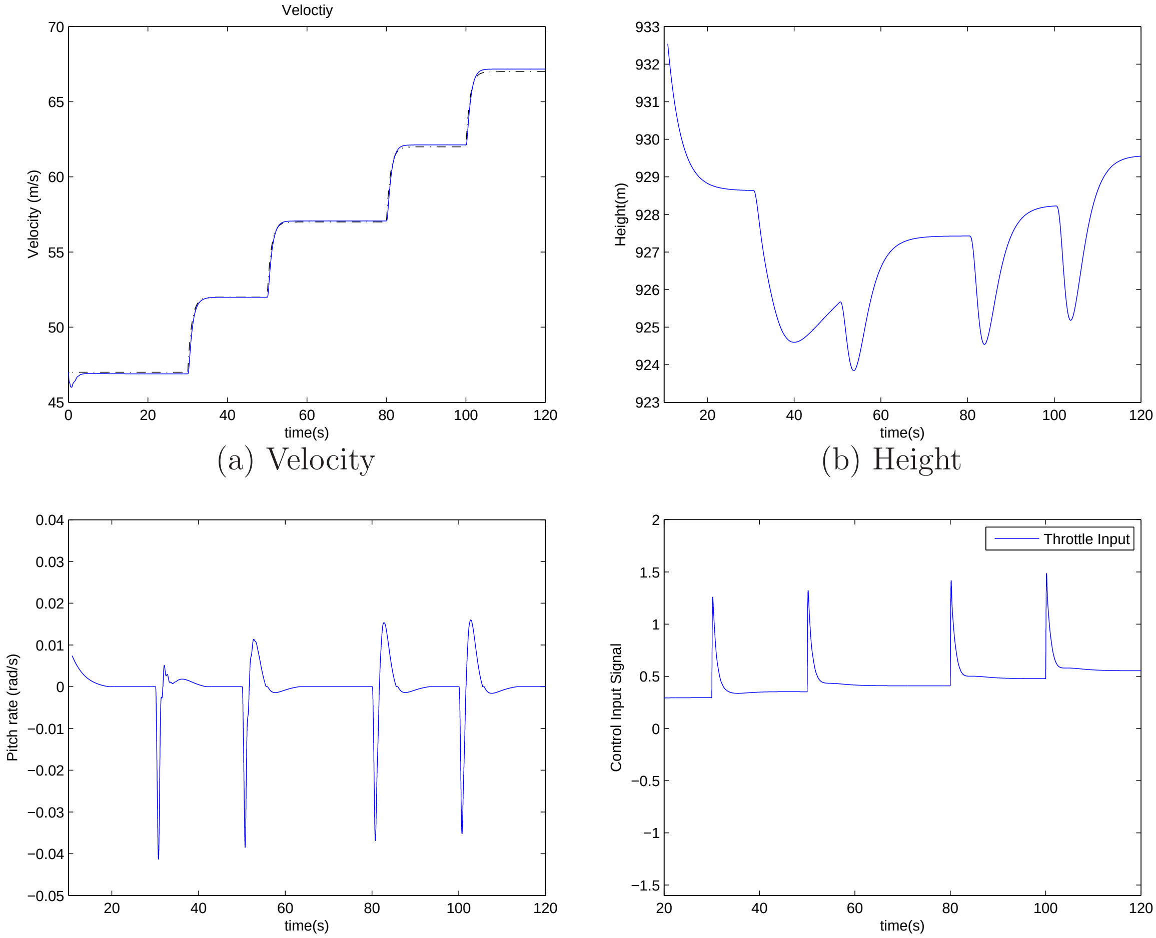
(a) Flight path angle

(b) Throttle/elevator inputs

(c) Height

(d) Velocity

Figure 6.11: Nonlinear simulation: tracking response of flight path angle for 2DOF controller



(a) Velocity

(b) Height

(c) Pitch rate

(d) Throttle inputs

Figure 6.12: Nonlinear simulation: tracking response of velocity for 2DoF controller

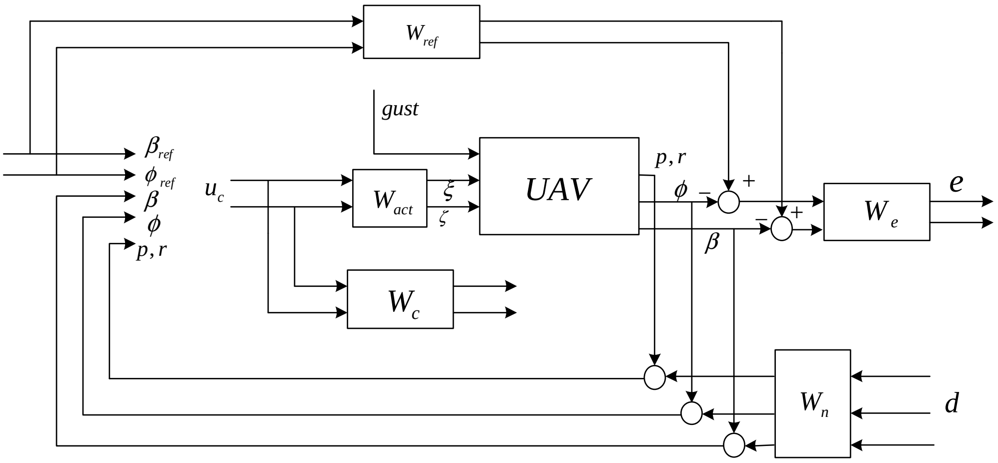


Figure 6.13: The lateral-directional flight control design plant

6.2.2 The Lateral-directional Control Design

The design plant of the lateral-directional system shown in Figure 6.13 largely resembles that of the longitudinal system. The output feedback measurements are sideslip angle (β), roll rate (p), yaw rate (r), roll angle (ϕ). The reference inputs are sideslip and roll angle. The control inputs (ξ, ζ) denote the aileron and rudder deflection. We only consider the lateral-directional dynamics here, the longitudinal dynamics are assumed to remain at equilibrium.

Performance objectives and weights

The performance goals of the lateral-directional closed-loop system based on the can be characterised as

- Maintain robust stability over the range $22 \leq V_t \leq 72$.
- Track unit step command of roll angle with rise time no greater than 5 seconds, overshoot no greater than 10% and steady state error no greater than 1%.
- Maximum elevator deflection and deflection rate do not exceed 20 deg and 35 deg/sec respectively.

The actuators for rudder and aileron are identical. The plant is augmented by the actuator and sensor models which are given by $W_{act}(s) = 20/(s+20)I_2$ and $50/(s+50)I_4$, respectively.

The ideal transfer functions W_{ref} for lateral control are given as

$$\begin{bmatrix} \psi_m \\ \beta_m \end{bmatrix} = \begin{bmatrix} \dfrac{0.7225}{s^2+1.7s+0.7225} & 0 \\ 0 & \dfrac{1}{s^2+2s+1} \end{bmatrix} \begin{bmatrix} \psi_{ref} \\ \beta_{ref} \end{bmatrix}$$

The guidelines for choosing the performance weights and control weights are similar to those for the longitudinal control design. We select the performance weight W_e as

$$W_e = \begin{bmatrix} \frac{0.1428s+14.28}{s+15} & 0 & 0 & 0 \\ 0 & \frac{0.1428s+14.28}{s+15} & 0 & 0 \\ 0 & 0 & \frac{0.2}{s+0.001} & 0 \\ 0 & 0 & 0 & \frac{30}{s+0.0001} \end{bmatrix},$$

and the control weights are chosen as

$$W_c = \begin{bmatrix} \dfrac{3.2s+28}{s+100} & 0 \\ 0 & \dfrac{8s+60}{s+400} \end{bmatrix}.$$

W_n is again chosen as an unit gain matrix. Here we

Computation and Synthesis

In summary, the 20th-order generalised plant has 4 output disturbances and 1 gust disturbance. There are 10 states in the UAV plant, sensors and actuator model. Basis functions for the Lyapunov matrices are selected as

$$X(\rho) = X_1, \;\; Y(\rho) = Y_1 + Y_2V_t + Y_3V_t^2.$$

We begin with the synthesis using constant performance level γ and the results are shown in Table 6.7. As expected, the PDLF synthesis is able to achieve better robust performance levels. Table 6.8 shows the level of robust performance for different parameterisations of the Lyapunov matrices. Again the quadratic parameterisation

Table 6.7: Comparison of SQLF and PDLF synthesis of lateral-directional control

	Number of LMIs	Number of dec.	LPV γ_{opt}	Comp. time
SQLF	23	421	51.87	1 min 29 secs
PDLF	44	841	29.90	7.25 mins

Table 6.8: Optimal LPV stability margin for different parameterisation of lateral-directional control

	Parameterisation	LPV optimal γ
I	$Y_0 + V_t Y_1$	31.28
II	$Y_0 + V_t^2 Y_1$	38.41
III	$Y_0 + V_t Y_1 + V_t^2 Y_2$	29.90
IV	X, Y constant	51.87

(III) has the lowest value of performance γ. The linear parameterisation (I) is also a better fit than (II).

Similarly, we define the parameterisation of γ as

$$\gamma(V_t) = \gamma_1 + \gamma_2 V_t.$$

The results are shown in the Table 6.9. Compared to $\gamma = 29.90$ of the synthesis with constant performance, the optimal $\mathcal{L}_2$ gains have been considerably reduced at lower velocities.

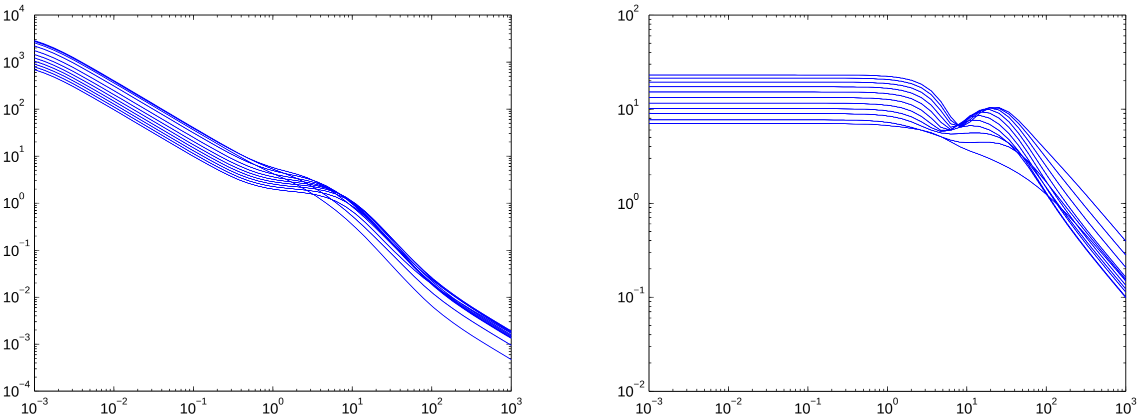
Table 6.9: γ at different grid points (SQLF)

V_t	22	27	32	37	42	47	52	57	62	67	72
γ	12.35	13.90	15.46	17.00	18.56	20.12	21.67	23.22	24.77	26.33	27.88

The original controller would be difficult to implement because one of the closed-loop poles was faster than 800 rad/s. This can be corrected using the pole placement approach. With trial and error, it is found that the fastest pole of the closed-loop system can be reduced to 100 rad/s. Table 6.10 shows the performance level with different pole placement constraints. We can see there is no significant degradation to the robust performance level. A frequency response of the lateral-directional controller is shown in Figure 6.14.

Table 6.10: Performance level γ vs. minimum pole locations constraints

Pole location	No constraints	-160	-130	-110
LPV γ_{opt}	29.90	30.09	30.12	30.23



roll angle reference to aileron

sideslip angle reference to rudder

Figure 6.14: Frequency response from reference to control input of lateral-directional controller

Simulation Results

We set the saturation limit for rudder as $[-0.4, 0.4]$ and aileron as $[-0.4, 0.4]$. The lateral-directional closed-loop responses are shown in Figure 6.15. We can see that the frequency responses from the reference command to outputs resemble those of the ideal reference model W_{ref}, which suggests good tracking performance. Figure 6.16 shows a flight test manoeuvre simulation for both the longitudinal and lateral-directional 2DoF/PDLF controllers. The manoeuvre starts from trimmed level flight at $(Vt, h) = (27m/s, 1000m)$, and consist of the following sequence of commands:

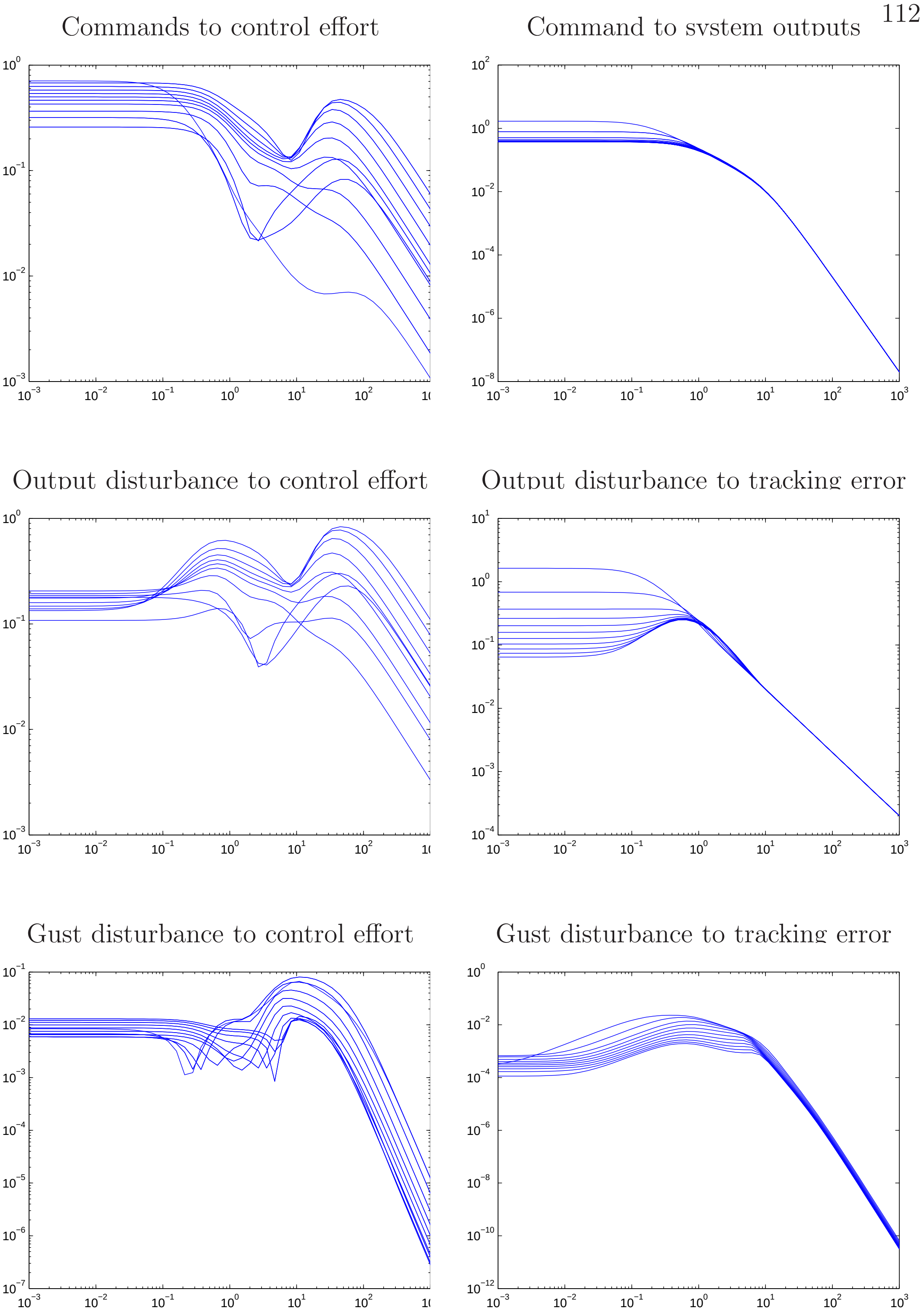
1. Roll right by 30 degrees for 30 seconds.

2. Pitch up by 10 degrees for 40 seconds, then pitch down to level flight.

3. Roll left by 0 degrees for 30 seconds.

4. Velocity remains steady then increases.

The piecewise steps that compose the command signals are prefiltered by $10/(s+10)$ to reflect the pilot bandwidth. Velocity is tracking a reference command that traverses the whole flight envelope in order to demonstrate the performance of the scheduling controller. We can see there are some cross couplings between the flight path angle and side slip channels, and some oscillations occur at the beginning of tracking of flight path angle because the initial states of the plant need to be adjusted to trim. We can see the response of flight path angle is slightly degraded due to saturation.



Commands to control effort

Command to system outputs

Output disturbance to control effort

Output disturbance to tracking error

Gust disturbance to control effort

Gust disturbance to tracking error

Figure 6.15: Lateral-directional closed-loop response by different grid points

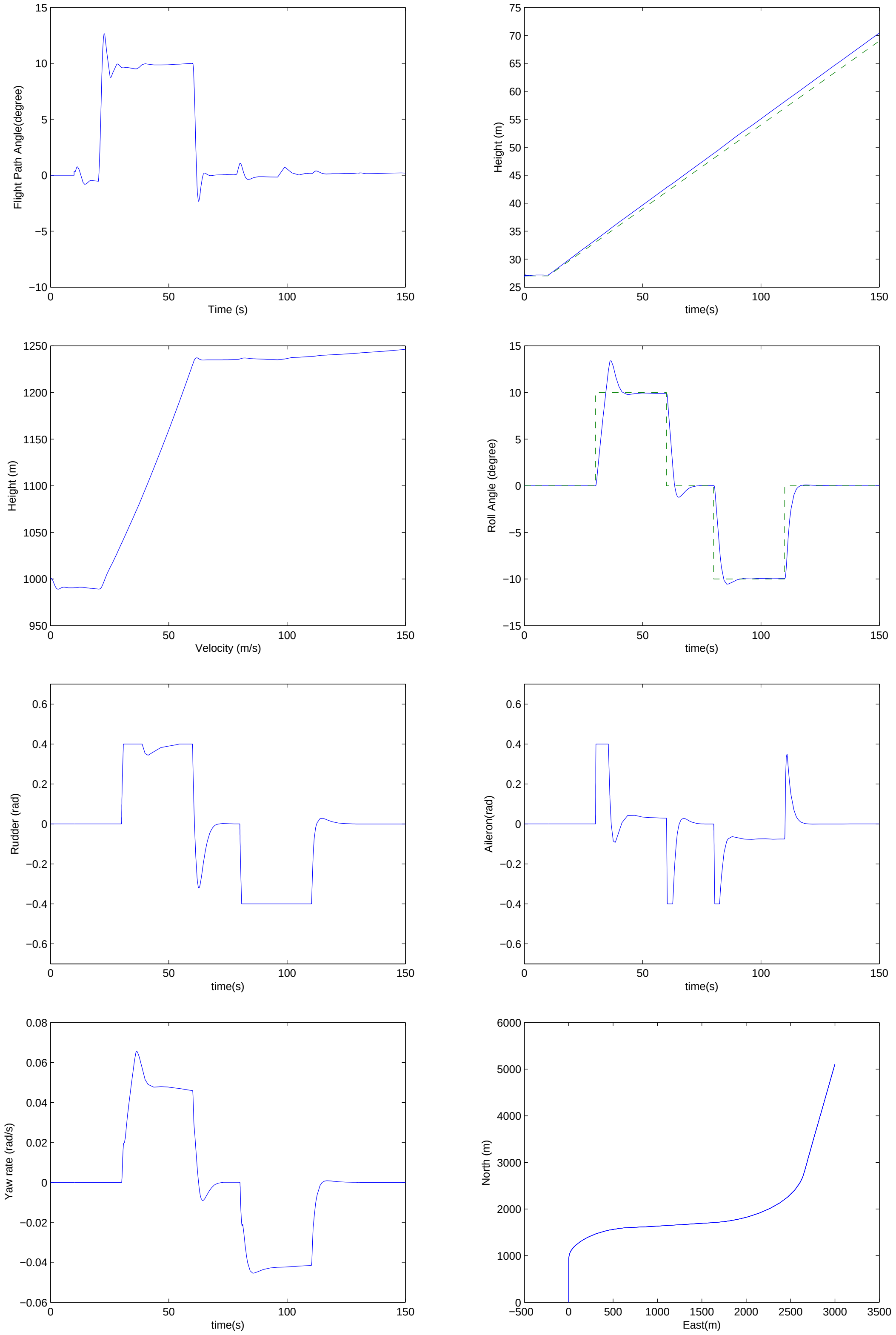


Figure 6.16: Combined nonlinear simulation in longitudinal and lateral manoeuvres (clockwise from the top-left: flight path angle, velocity, roll angle, aileron, manoeuvre in the north-east coordinate, yaw rate, rudder, height

6.3 The Loop shaping/PDLF Design

In this section, we will design and simulate longitudinal and lateral LPV controllers for the UAV using the loop shaping/PDLF method that we proposed in §4.3.

6.3.1 The Longitudinal Control Design

The LPV model of the plant is augmented by first order approximations of actuator dynamics given as

$$W_{act} = \begin{bmatrix} \dfrac{5}{s+5} & 0 \\ 0 & \dfrac{20}{s+20} \end{bmatrix}$$

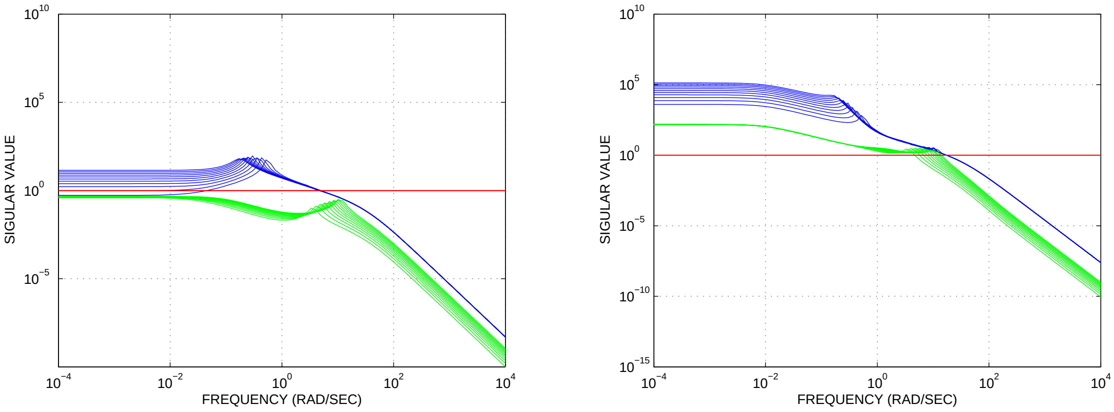
and the sensor dynamics are denoted by $\dfrac{50}{s+50} I_4$. In the loop shaping design, the next step is to shape the open-loop plant. Pre-compensators containing "integrators" are selected for both the throttle and elevator to boost the low frequency gain and reduce the zero steady-state error. The pre-compensators are given as

$$W_1 = \begin{bmatrix} k_{11}\frac{0.009s+1}{s+0.01} & 0 \\ 0 & k_{12}\frac{s+0.6}{s+0.01} \end{bmatrix} \tag{6.3}$$

where $k_{11} = 40$ and $k_{12} = 4$. To limit the plant states, a non-dynamic post-compensator is chosen as $W_2 = \text{diag}(0.8, 1.2, 1.1)$ to emphasise the outputs to be controlled. An open-loop frequency response of the original plant and the shaped plant (given as $G_s = W_2 G(\rho) W_1$) can be seen in Figure 6.17. $G_s(\rho)$ has only 11 states which is significantly less than the 2DoF design, and includes 4 states from the original model, 3 states from the sensors, 2 states from the actuators and 2 states from the pre-compensator.

Controller Synthesis

The LPV loop shaping control design structure was shown in Figure 4.2 in §4.2. We derive the left coprime factorisation of the augmented plant by Theorem 4 in §4.2.1, and the optimisation yields matrices P and Z. The global minimum of the objective trace(Z) = 1256.7. Using the formula $L = -P^{-1}C^T$, we have L given as



(a) Original plant

(b) Shaped plant

Figure 6.17: Open-loop responses of the lateral-directional plant

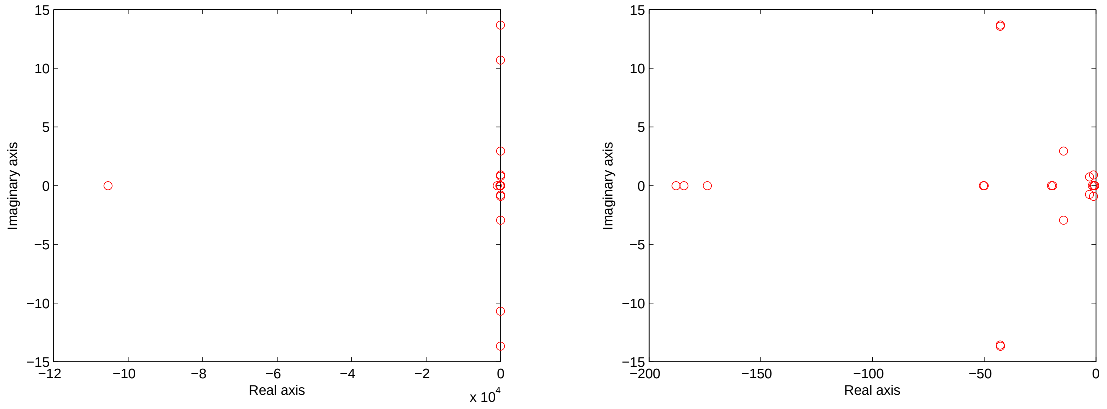
an 11 by 2 matrix. The synthesis procedure is to solve the three LMIs in Theorem 6 of §4.2.2. To apply the PDLF synthesis, the dependence of Lyapunov matrices X, Y on the scheduling parameter has to be decided. Based on the discussion given in the previous chapter, the following second order parameterisation is chosen:

$$Y = Y_0 + V_t Y_1 + V_t^2 Y_2.$$

To avoid introducing derivative terms in the controller construction, X is selected as a constant matrix. The rate of variation is set to $-10 \leq \dot{V}_t \leq 10$. The optimisation procedure yields $\gamma = 17.45$. For comparison, the procedure that uses the SQLF method was also calculated with the identical setup, and we got $\gamma = 36.27$. The detailed results can be seen in Table 6.11. We can construct the controller using the formula in (4.35). The final controller is given by $W_2 K(\rho) W_1$ and is 13th order.

Table 6.11: Comparison of SQLF and PDLF longitudinal loop shaping design

	Number of LMIs	Number of dec.	LPV γ_{opt}	Comp. time
SQLF	23	133	36.27	6.8 seconds
PDLF	44	265	17.45	22 seconds



(a) before

(b) after

Figure 6.18: Comparison of the pole locations for loopshaping/PDLF method before and after adding the constraints

The Pole Placement Approach

The longitudinal controller synthesised by the loop shaping/PDLF approach has one fast closed-loop pole. Therefore we add extra pole placement constraints into the LMI synthesis. The limits of the closed-loop poles are investigated by experiment and the results are shown in the Table 6.12. The minimum closed-loop pole is larger than -190 with the vertical strip convex region $\mathcal{D} = \{x+jy \in \mathbf{C} : \ -190 < x < -0.5 < 0\}$. The pole locations before and after adding the pole placement constraints can be seen in Figure 6.18.

Table 6.12: The poles of the longitudinal closed-loop system with the loop shaping/PDLF controller

Poles constraints	minimum poles	γ
no constraints	-1×10^5	16.64
pole > -1000	-995.96	17.45
pole > -500	-497.34	17.51
pole > -220	-217.75	17.67
pole > -190	-187.62	17.71

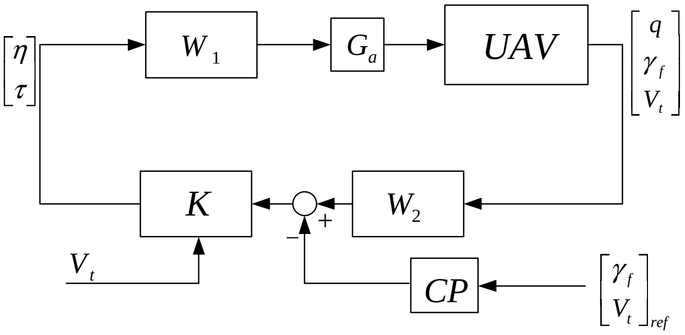

$\begin{bmatrix}\eta\\ \tau\end{bmatrix}$
W_1
G_a
UAV
$\begin{bmatrix}q\\ \gamma_f\\ V_t\end{bmatrix}$
K
+
−
W_2
V_t
CP
$\begin{bmatrix}\gamma_f\\ V_t\end{bmatrix}_{ref}$

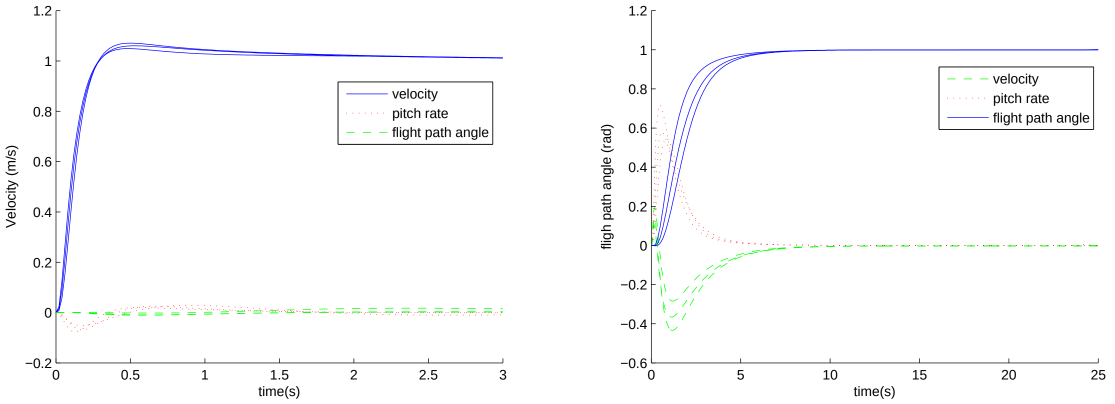
Figure 6.19: The longitudinal closed-loop configuration for the loop shaping controller

Simulation Results

The closed-loop feedback system is illustrated in Figure 6.19. The controller K has been implemented in the forward path, and CP is a command prefilter. The saturation limits for throttle and elevator are set as $[-2, 2]$ and $[-15, 15]$ respectively. Figure 6.20 shows the step demand responses of velocity and flight path angle about different airspeeds. The responses are fast and well damped with little steady state error. We then implement the LPV controller in a nonlinear parameter varying simulation. Figure 6.21 shows the tracking control of flight path angle to a doublet $[-30°, 30°]$ reference input. The velocity channel is tracking a reference steadily climbing from 22m/s to 72m/s and the controller is scheduled in real-time. We can see good tracking accuracy of velocity, while the response of flight path angle has a little overshoot which is a common problem with loop shaping designs. Figure 6.22 shows the tracking of velocity in response to multiple steps. Again, the tracking of velocity exhibits more overshoot than that of the 2DoF/PDLF design.

Uncertainty Analysis

It is important for the flight control system to be robust to changes in the flight conditions. The closed-loop system behaviour should not be affected too much by actuator uncertainties, noise measurements and parametric uncertainties. As the



(a) Velocity

(b) Flight Path Angle

Figure 6.20: Velocity and flight path angle step response at 22, 47, 72m/s

loop shaping synthesis implies robust stabilisation against stable structural, comprime factor uncertainties, we will investigate the effects of uncertainties in the aerodynamic data on UAV flight performance.

For light-weight aircraft such as UAVs, variations in the payload will have significant impact on the dynamics of the system. Therefore, we will consider the variations of mass (M_s) as well as centre of gravity (X_{cg}) which denotes the position of centre of gravity from the nose of the aircraft.

By nonlinear simulation, we found the bounds such that the flight control system can maintain robust stability:

$$0.4M_s < M_s' < 3.2M_s$$
$$0.4xcg < X_{cg}' < 1.05X_{cg}$$

When the variations of the parameters exceed the bounds, the simulation results show that the system either loses robust stability or performs poorly. It is interesting to see that the control system can tolerate large variations in mass, but with very little tolerance of pulling-back the centre of gravity. Figure 6.23 shows the tracking performance of the flight path angle under some of the worst cases. Here G denotes the nominal plant, G_1 denotes the plant with mass and centre of gravity $[M_s', X_{cg}'] =$

$[0.4M_s, 0.4X_{cg}]$, and G_2 with $[3.2M_s, 1.05X_{cg}]$. Velocity is tracking a ramp climbing from 22m/s to 72m/s during the simulation.

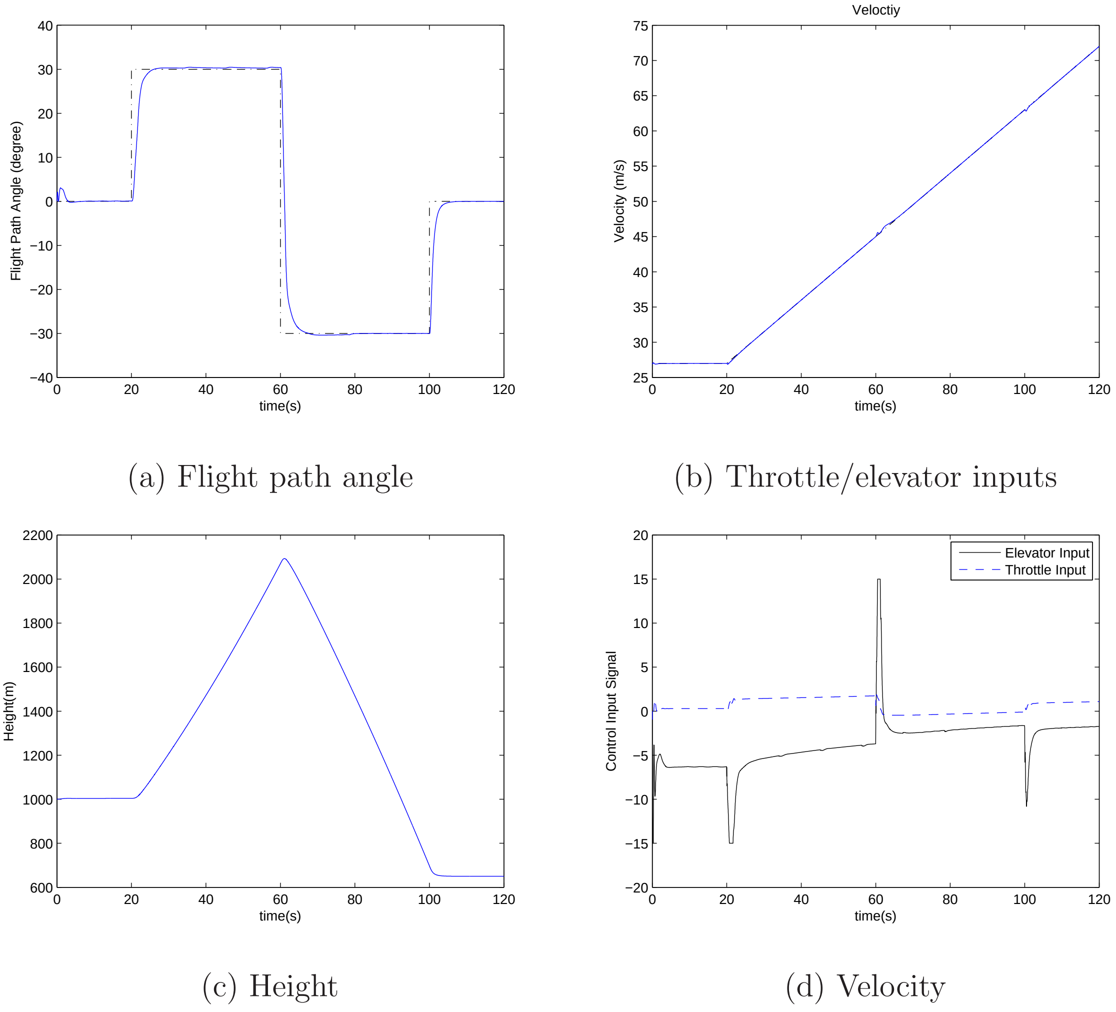
The flight control system survives the perturbed plants G_1 and G_2, but there is degradation in the tracking response. This is particularly so for the G_1 type of perturbation which causes large amounts of overshoot.

We will also investigate how the aerodynamics data such as pitch moment coefficient (C_M) and lift coefficient (C_L) affect the controller performance. These coefficients might be inaccurate due to modelling errors. The bounds of the stability region are found by trial and error in the nonlinear simulation:

$$0.2C_L < C'_L < 10C_L$$
$$0.7C_M < C'_M < 2.1C_M$$

Figure 6.24 shows two worst case tracking simulation results with nominal models G, and two perturbed model G_1 and G_2, where G_1 denotes the perturbed condition with uncertainties $[0.7C_M, 10C_L]$, and G_2 is given by $[0.2C_L, 2.1C_M]$. We can see that tracking performance of the flight path angle is more vulnerable to the variations of the pitching moment coefficient as expected from their aerodynamic relationship.

From the simulations, we are confident that the loop shaping/PDLF controller has some amount of tolerance to variations in payloads and flight conditions.



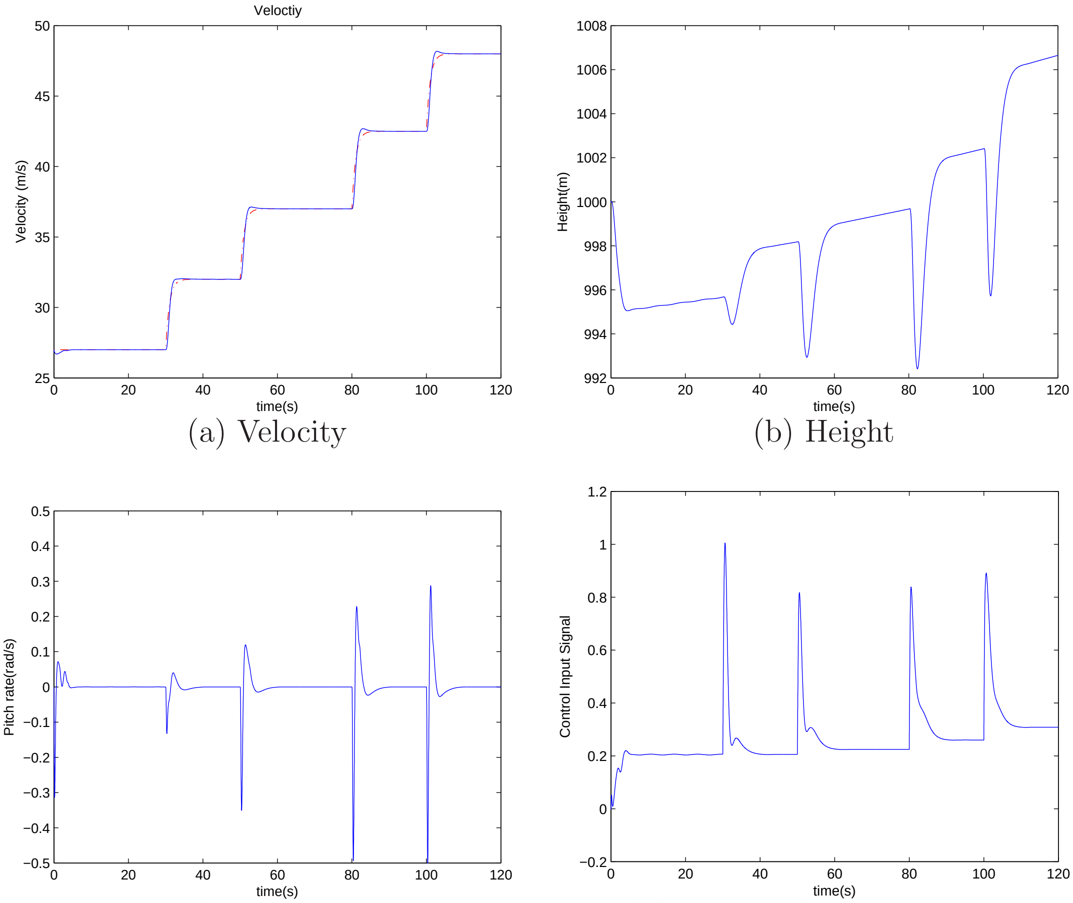
(a) Flight path angle

(b) Throttle/elevator inputs

(c) Height

(d) Velocity

Figure 6.21: Nonlinear simulation: tracking response of flight path angle for the loop shaping controller



(a) Velocity

(b) Height

(c) Pitch rate

(d) Throttle/elevator inputs

Figure 6.22: Nonlinear simulation: tracking response of velocity for the loop shaping controller

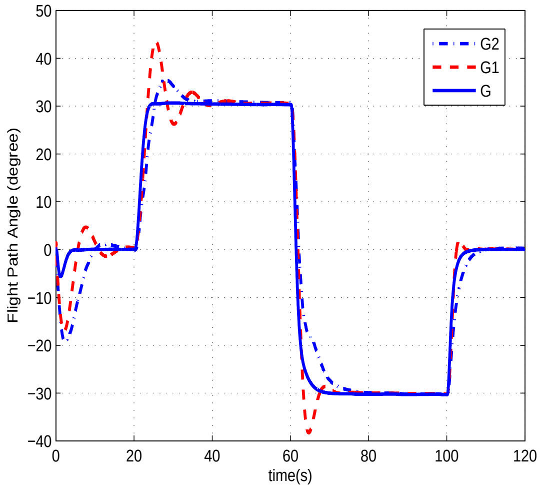


Figure 6.23: Worst case tracking performance for Mass/Cg variation

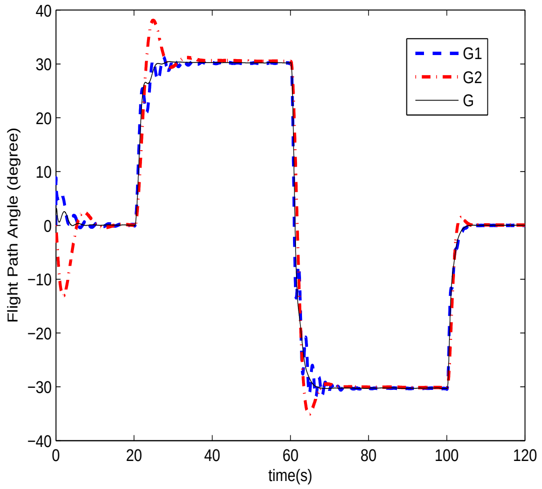


Figure 6.24: Worst case tracking performance for C_L/C_M variations

6.3.2 The Lateral-directional Control Design

The lateral-directional loop shaping control design will have the same closed-loop configuration as the longitudinal control design. The pre-compensators are chosen as

$$W_1 = \mathrm{diag}\left(\mathrm{k}_{11}\frac{\mathrm{s}+0.01}{\mathrm{s}+0.0001}, \mathrm{k}_{12}\frac{\mathrm{s}+0.01}{\mathrm{s}+0.0001}\right)$$

where $k_{11} = 1.5$ and $k_{12} = 4$. Again the integral parts are used to boost the low frequency gain and improve performance. The post-compensator has been chosen as unit gain in this case.

Computation and synthesis

We derive the left coprime factorisation of the augmented plant using the LMIs in Theorem 4. The global minimum of the trace of Z is trace(Z) = 1256.7. Using the formula $L = -P^{-1}C^T$, L is given as an 11 by 4 matrix. The synthesis procedure yields $\gamma = 11.18$. Results for comparison can be seen in Table 6.13. The final controller is augmented by pre and post-compensators and is 13th order. The fastest pole of the closed-loop system is found to be 49.5 rad/s, and therefore no pole placement constraints are needed.

Table 6.13: Comparison of SQLF and PDLF longitudinal loop shaping design

	Number of LMIs	Number of dec.	LPV γ_{opt}	Comp. time
SQLF	23	133	14.85	5.4 seconds
PDLF	44	313	11.18	27 seconds

Simulation Results

The implemented closed-loop system is illustrated in Figure 6.25, with the controller K implemented in the forward path. Figure 6.26 shows the combined simultaneous simulation of longitudinal and lateral-directional loop shaping/PDLF controllers. The manoeuvre starts from trimmed level flight at $(Vt, h) = (32m/s, 1000m)$, and consist of the following sequence of commands:

1. Roll right by 15 degrees for 30 seconds.

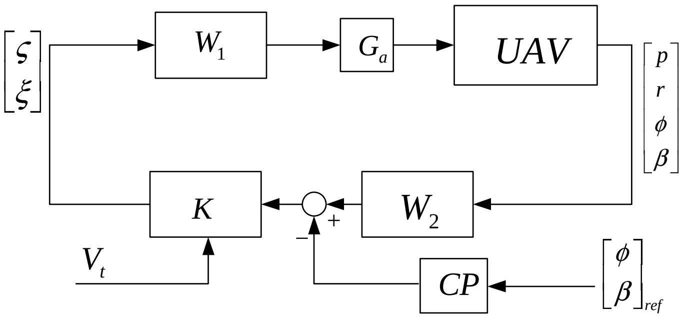

$\begin{bmatrix} \varsigma \\ \xi \end{bmatrix}$
W_1
G_a
UAV
$\begin{bmatrix} p \\ r \\ \phi \\ \beta \end{bmatrix}$
K
W_2
+
−
V_t
CP
$\begin{bmatrix} \phi \\ \beta \end{bmatrix}_{ref}$

Figure 6.25: The lateral closed-loop configuration for loop shaping controller

2. Pitch up by 10 degrees for 40 seconds, then pitch down to level flight.

3. Roll left by 30 degrees for 30 seconds.

4. Velocity remain steady, then increases.

The piecewise steps that compose the command signals are prefiltered by $CP = 10/(s+10)I_2$ to reflect better the pilot bandwidth. The velocity traverses the whole flight envelope in order to demonstrate the performance of the gain-scheduled controller.

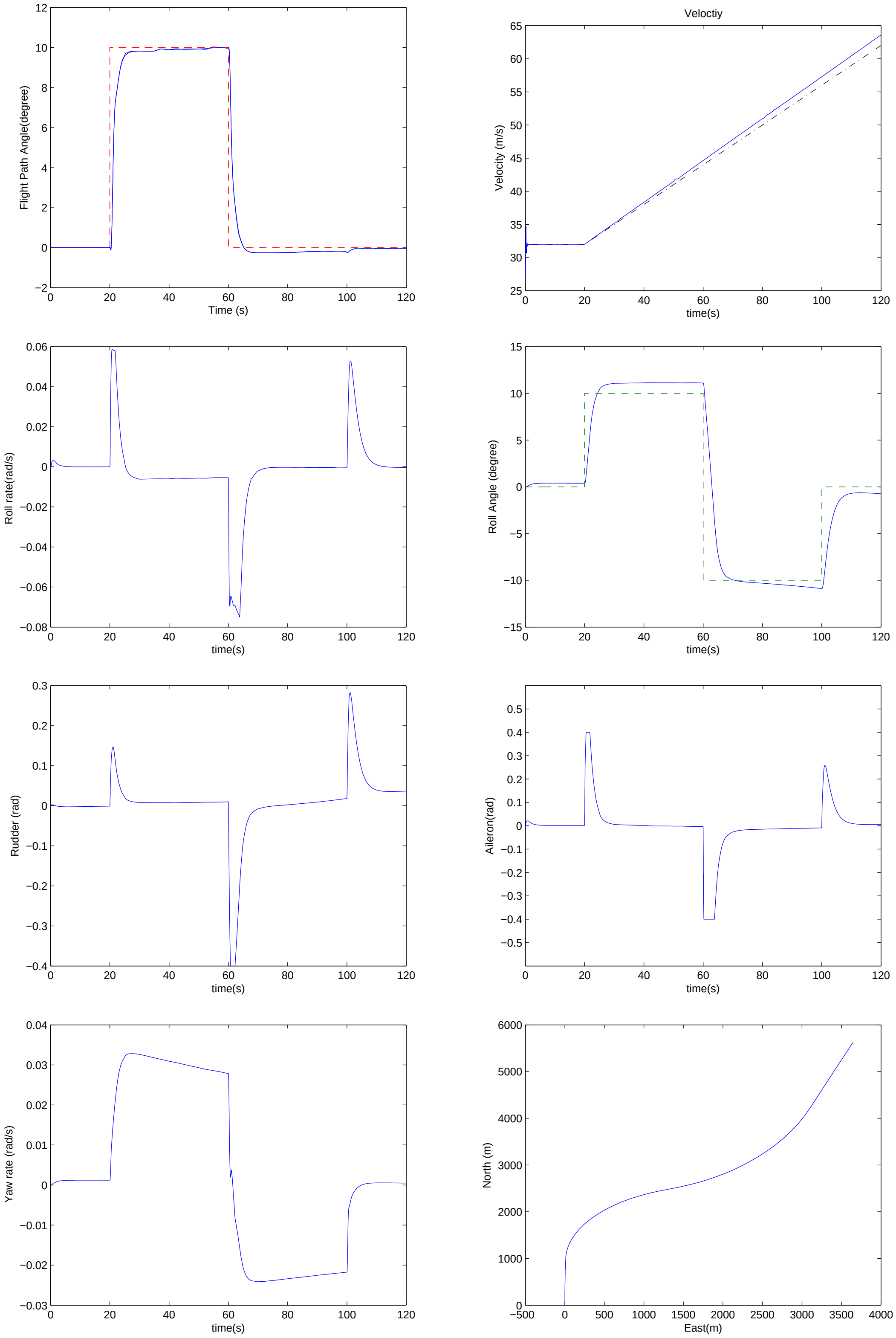


Figure 6.26: Combined simulation: (clockwise from top left) Flight path angle, Velocity, Roll angle, Aileron, North-east coordinate manoeuvre, Yaw rate, Elevator, Roll rate

6.4 Chapter Summary

In this chapter, we have designed the 2DoF/PDLF and loop shaping/PDLF controllers separately for the longitudinal and lateral dynamics of the UAV. In the controller synthesis process, we are able to demonstrate the results and provide evidence to answer the questions given at the beginning of this chapter. Comparing these two design methods, we have the following comments:

- The 2DoF/PDLF method tends to have a more complex closed-loop structure, resulting in more computational complexity and greater computation time.

- The 2DoF/PDLF method uses a mixed-sensitivity approach, so we are able to specify directly the performance of the system against exogenous disturbances at the control inputs and system outputs.

- The 2DoF/PDLF controller has higher order than loop shaping/PDLF controller.

- The Loop shaping/PDLF design as an extension of the well known H_∞ loop shaping methodology, is able to address robustness in the presence of stable coprime factor uncertainties.

- Both methods have fast response and little steady state error. In general, the loop shaping/PDLF controller seems to exhibit more overshoot, but less cross channel couplings, when compared to the 2DoF/PDLF controller.

The nonlinear simulations of the controller are given, the performance and stability of the gain-scheduling controller were evaluated with full flight envelope conditions. The simulation results show satisfactory tracking response and robust performance.

Chapter 7

Conclusions

In this thesis, we have reviewed linear parameter varying controller design using parameter dependent Lyapunov functions. With reference to the work done by other researchers [5, 15, 24, 34, 42], we have proposed two design methods: 2DoF/PDLF and loop shaping/PDLF. The design procedures are also presented. The solvability conditions of these methods are reduced to parameter dependent LMIs which can be solved efficiently using convex optimisation techniques. They are particularly suited to the design of gain-scheduled flight control systems, as they can deal systematically with time-varying, parameter dependent MIMO systems with model uncertainties and environmental perturbations. Moreover, the designers can build closed-loop performance specifications into the controller synthesis. In the sequel, we will summarise and comment on the main contributions of the thesis.

We have reviewed the notion of robust stability and parameter dependent performance of LPV systems and the synthesis of γ performance output feedback controllers. We have discussed the practical issues of the computation and implementation for the PDLF synthesis.

We have formulated and solve the LPV control problem with parameter dependent performance index. A 2DoF control structure is used in the output feedback synthesis. A varying performance index allowed the closed-loop system to have different levels of performance with different operating points, which could be implemented naturally in a gain scheduled controller. In contrast to the IQC method, we proposed a parameterisation method with the functional dependence of γ copying the plant nonlinearity. This parameterisation method contributes to a more flexible and efficient generalisation of the LMI synthesis.

The H_∞ loop shaping paradigm was adopted in the PDLF synthesis. We derived the left coprime factorisation using a robust H_2 filtering technique. We changed the original assumptions about the LPV plant structure and proved that the results can still hold with this change. Subsequently the LMIs for solving the loop shaping/PDLF output feedback problem were derived and explicit controller formulas were given. To our knowledge, this is the first implementation of LPV loop shaping controller design using Wu's PDLF methods in a coprime factorisation design framework.

A practical consideration of the closed-loop pole locations motivated us to incorporate the pole placement constraints into the PDLF synthesis. The idea of an LMI region was used. We defined a vertical strip region and showed that the closed-loop poles of the output feedback controller design could be forced into such a region by adding some LMI constraints. In chapter 6, we added the pole placement LMI constraints to the two different control synthesis procedures presented in this thesis. The results showed that the closed-loop poles can be placed in the pre-specified region and the fast mode of the controller can be eliminated. Furthermore, we investigated the controller responses to some aggressive step demands and it was seen that the controller dynamics were substantially improved.

In order to apply the LPV control techniques, we converted the nonlinear model of the UAV into an LPV form by using Jacobian linearisations around a family of equilibrium points. The LPV model was then validated by comparing time responses with those of the nonlinear model. The simulation results showed that the LPV model can capture the characteristics of the nonlinear model around most of the equilibria. We also analysed the dynamics of the LPV model. By obtaining an affine parameter dependent model of the UAV, the parameter dependence of the nonlinear model was investigated, and the results were used in the controller design phase.

We applied the controller design methods developed in this thesis to the synthesis of longitudinal and lateral flight controllers for the "Eclipse" flying UAV demonstrator. The controllers were synthesised using both SQLF and PDLF approaches.

The results indicated that the PDLF approach offers less conservative solutions for the controller synthesis without significantly increasing the problem complexity. A comparison of the synthesis results of different parameterisations for the Lyapunov matrices justifies the rule of choosing the parameter dependence. We showed that better robust performance was achievable at lower velocities if the problems are synthesised with parameter dependent performance. We studied the effects of variations of aerodynamic coefficients on the loop shaping/PDLF controller, and the results indicate that the closed-loop system can tolerate some amount of uncertainties. The nonlinear time simulations for both longitudinal and lateral manoeuvres showed satisfactory flight performance.

In addition to the results presented in this thesis, there are some unsolved problems which provide useful future directions of research:

- In the thesis, we have assumed that the measured parameters are accurate. However, this is not true with the presence of sensor noise and time delays. A time delay makes the measurement not "real-time", especially when the time delay depends on parameter trajectories. How to cope with a parameter varying time delay in our current LPV design framework is an open problem for research.

- In our design example, we only have one design parameter which poses no burden on computational complexity. However, the computation time tends to increase exponentially with an increase in the number of parameters and grid points. In recent years, the so-called randomised approach has emerged as a complementary tool for handling uncertain systems. Similar to the Monte Carlo method, the complexity of the randomised approach will not be affected by the dimensions of the problem. It has now been used in the probabilistic LPV control of an F-16 aircraft [62]. Therefore it would be interesting to implement the randomised approach in the PDLF design framework.

- We did not consider the problems of actuator saturation in the design example. A well-designed, linear parameter varying anit-windup compensator could be applied to the control loop to cope with the saturation problems. This presents a potential area for research.

- A static feedback controller can be designed using the PDLF synthesis with some modification of the assumptions. It would be worth investigating if a static controller can be synthesised for this UAV design example.
- Finally, further work can be done to assess the benefits of implementing the loop shaping/PDLF controller in an observer form.

Appendix

Geometric configurations and aerodynamic data of the "Eclipse" UAV

Wing Geometry

Table 7.1: Wing geometry

Name	Symbol	Value
Wing Area (m^2)	$\mathbf{S}$	2.365
Span (m)	b	2.2
Wing mean chord (m)	c_{bar}	1.34
Aspect Ratio	$\mathbf{AR}$	2.047

Fin Geometry

Table 7.2: Fin geometry

Name	Symbol	Value
Fin Area (m^2)	$\mathbf{S}_F$	0.137
Span (m)	h_F	0.425
Fin mean chord (m)	c_{bar_f}	0.375
Aspect Ratio	$\mathbf{AR}_F$	1.32

Control Surfaces

The control surface configuration has 2 trailing edge devices on either side of the certre line. The trailing edge devices were divided in 2 areas:

- A: aileron type device
- D: elevator type device
- No flap

The TEDs surface have been extended 1.5 times chord wise.

Mass and Centre of Gravity Location

The flying demonstrator UAV centre of gravity position has been shifted slightly forwards, measured with respect to the nose point its flat base.

Table 7.3: Mass and centre of gravity location

Name	Symbol	Value
Mass (kg)	M_s	39.7
C.G. position(X axis) (m)	X_{cg}	1.217
C.G. position(Y axis) (m)	Y_{cg}	0
C.G. position(Z axis) (m)	Z_{cg}	0.034

Aerodynamics data

Table 7.4: Moment of inertia in body axe

Name	Symbol	Value
Mass (kg)	M_s	39.7
C.G. roll inertia (kgm^2)	I_x	1.5
C.G. pitch inertia (kgm^2)	I_y	10.43
C.G. yaw inertia (kgm^2)	I_z	11.41
C.G. product inertia (kgm^2)	I_{xy}	0
C.G. product inertia (kgm^2)	I_{xz}	-0.14
C.G. yaw inertia (kgm^2)	I_{yz}	0

Table 7.5: Longitudinal derivatives

Name	Symbol	Value
Lift curve slop	C_{L_α}	2.51
Elevator lift coefficient	C_{L_η}	0.157
Pitch damping moment coefficient	C_{M_q}	0.564
Zero lift drag coefficient	C_{D_0}	0.0172

Table 7.6: Lateral derivatives

Symbols	Value	Symbols	Value
C_{Y_ζ}	0.14	CY_β	-0.086
C_{Y_r}	0.120	C_{Y_p}	-0.221
C_{LL_ζ}	0.120	C_{LL_β}	-0.031
C_{LL_r}	0.097	C_{LL_p}	0.164
C_{N_ζ}	-0.403	C_{N_β}	0.421
C_{N_r}	0.030	C_{N_p}	-0.027

Flow chart of the synthesis procedure

The design cycle flow chart can be seen from figure 7.1

Source code of the controller synthesis

Closed-loop construction for 2DOF

```
clear
% Load Plant Model for Given Velolcity..
Vt = [22    27    32    37    42    47    52    57    62    67    72];
load Along.mat;load Blong.mat;load Clong.mat;load Dlong.mat;load alpt.mat
% Consturcting varying system matrix
for i=1:11
An = Along(:,:,i);
Bn1 = Blong(:,:,i);
C = [0 0 0 1; 0 -1/Vt(i) 1 0;cos(alpt(i)) sin(alpt(i)) 0 0];
D = zeros(3,2);
```

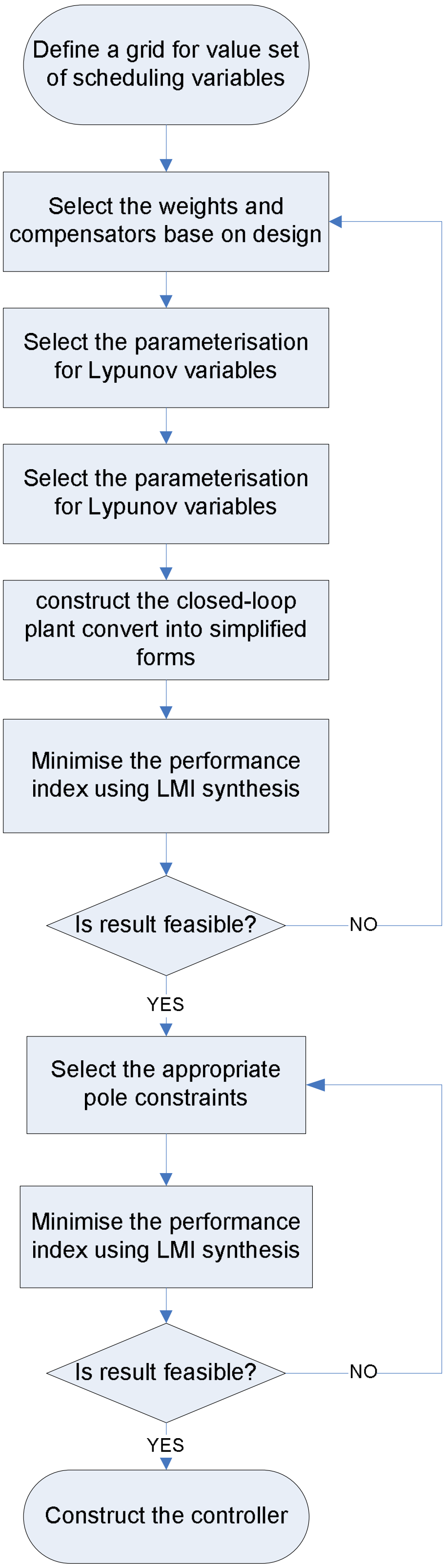

Define a grid for value set of scheduling variables
Select the weights and compensators base on design
Select the parameterisation for Lypunov variables
Select the parameterisation for Lypunov variables
construct the closed-loop plant convert into simplified forms
Minimise the performance index using LMI synthesis
Is result feasible?
NO
YES
Select the appropriate pole constraints
Minimise the performance index using LMI synthesis
Is result feasible?
NO
YES
Construct the controller

Figure 7.1: Design cycle for PDLF synthesis

```
%Adding Gust model
Bgust = -[An(1,2);An(2,2);An(3,2);An(4,2)];
Bn2=[Bgust Bn1];
D = [zeros(3,1) D];
%-----scaling-------------
in=[1.2 0; 0 .8 ]
out=[1 0 0; 0 1.2 0; 0 0 0.8]
% %----------------------
ss1 = ss(An,Bn1,C,D(:,2:3));
ss2 = ss(An,Bn2,C,D);
na = [20];
da = [1 20];
tac1=tf(na,da);
tac2=tf(5,[1 5]);
tacs=[ tac1 0;0 tac2];
tacss=ss(tacs);
sys = series(tacss,ss1,[1,2],[1,2]);
tsenvel = tf(50,[1 50]);
tsenz = tf(50,[1 50 ]);
tsenq = tf(50,[1 50]);
tsens = [tsenq 0 0;0 tsenz 0;0 0 tsenvel];% 0;0 0 0 tsenvel];
[ass,bss,css,dss] = ssdata(tsens);
sysnew = series(sys,tsens);
outss=ss(out);
sysnew=series(sysnew, outss);
%-- Input (r)  sensitive weighting function-----------
k = 4; omeg = 20;
numw2 = [0.2 1];
denw2 = [1 k*omeg];
numw21 = [0.2 1];
denw21 = [1 k*omeg];
```

```
w21 = tf(numw2,denw2);
w22 = tf(numw21,denw21);
% w21=tf([10 200], [1 2000])
% w22=tf([10 200], [1 2000])

%-- output (z) sensitive weighting function----------
 om = 10;
 ze = 1;
denwd11= [1 2*ze*om 1];
numwd11= 1*[100 1];
w11=tf(numwd11,denwd11);
w12=tf(15*[0.01 1],[0.07 1])
w12=tf(1,[1 1e-4]);
numwd13= 0.1428*[1 1000];
denwd13= [1 15];
w13=tf(numwd13,denwd13);
output_weight= [w11 0 0; 0 w12 0; 0 0 w13];
outss=ss(out);
out_weightt=series(output_weight,outss );
%---ideal model----
wm1=tf(100,[1 20 100])
wm2=tf(100,[1 20 100])
%------Setting up matrices for augmentation-------
 [a,b,c,d] = ssdata(sysnew);
 [a1,b1,c1,d1]=ssdata(out_weightt);
 [a2,b2,c2,d2]=ssdata([w21 0; 0 w22]);
 [am,bm,cm,dm]=ssdata([wm1 0; 0 wm2]);
sysm=pck(am,bm,cm,dm);
%getting size of the matrix                                                 %
 [ar,ac]= size(a);[br,bc]=size(b);[cr,cc]=size(c);[dr,dc]=size(d);
 [a1r,a1c]=size(a1);[b1r,b1c]=size(b1);[c1r,c1c]=size(c1);[d1r,d1c]=size(d1
[a2r,a2c]=size(a2);[b2r,b2c]=size(b2);[c2r,c2c]=size(c2);[d2r,d2c]=size(d2
[amr,amc]=size(am);[bmr,bmc]=size(bm);[cmr,cmc]=size(cm);[dmr,dmc]=size(dm
% gust model
bg=[zeros(3,1);Bgust;zeros(2,1)];
dg=zeros(3,1);
%Augmenting the system matrices [A Bd Bu; Ce Ded Deu; Cy Dyd Dyu] /
%e=[e1;e2]  Ce=[Ce1; Ce2] De=[Ded Deu]
A= [a             zeros(ar,a1c)   zeros(ar,a2c)    zeros(ar,amc);   %  X
   b1*c            a1             zeros(a1r,a2c)   -b1*[zeros(1,4);cm];
   zeros(a2r,ac)  zeros(a2r,a1c)  a2               zeros(a2r, amc) ;  %  X
   zeros(amr,ac)  zeros(amr,a1c)  zeros(amr,a2c)   am;              %  Xm
   ]

B=[bg             zeros(br,bmc)*in     zeros(br,b1c)    b ;
   b1*dg          (-b1*[zeros(1,2);dm])*in   b1          b1*d;
   zeros(b2r,1)   (zeros(b2r,bmc))*in    zeros(b2r,b1c)   b2;
   zeros(bmr,1)   bm*in                zeros(bmr,b1c)   zeros(bmr,bc);]

z1=3;z2=2;e1=2;e2=3;  %no. of input/output

 C1=[d1*c            c1             zeros(c1r,c2c) -d1*[zeros(1,4);cm];
     zeros(c2r,cc)   zeros(c2r,c1c)  c2             zeros(c2r,cmc);]

 C2=[zeros(e1,cc)    zeros(e1,c1c)   zeros(e1,c2c)  zeros(e1,cmc);
         c           zeros(cr,c1c)   zeros(cr,c2c)  zeros(cr,cmc);]
 C=[C1;C2]

 D11=[d1*dg       (-d1*[zeros(1,2); dm])*in d1;
      zeros(d2r,1)  zeros(d2r,d2c)*in zeros(d2r,d1c)];
 D12=[d1*d;d2]
 D21=[zeros(e1,1)      eye(e1,dmc)*in zeros(e1,d1c);
```

```
      dg               zeros(cr,dmc)*in eye(cr,d1c);]
 D22=[zeros(e1, d2c); d]

 D=[D11 D12; D21 D22]
lpvsys(:,:,i)=pck(A,B,C,D);
end

vlpv=[];
vsysnew=[];
for i=1:11
    vlpv=[vlpv ; lpvsys(:,:,i)]
%    vsysnew=[vsysnew;sysnew1(:,:,i)];
end
iv=1:11
vlpv=vpck(vlpv,iv);
Vt=Vt'
viv=vpck(Vt,iv);
vsysnew=vpck(vsysnew,iv)
```

Closed-loop construction for loop shaping

```

load Along.mat;load Blong.mat;load Clong.mat;load Dlong.mat;load alpt.mat
Vel=[22    27    32    37    42    47    52    57    62    67    72];
% Load Plant Model for a Given Velolcity..#
om=logspace(-4,4,100)
for i=1:11
An =Along(:,:,i);
Bn =Blong(:,:,i);
C = [0 0 0 1; 0 -1/Vel(i) 1 0;cos(alpt(i)) sin(alpt(i)) 0 0];
D = zeros(3,2);
```

```
ss2 = ss(An,Bn,C,D);
na = [20];
da = [1 20];
tac=tf(na,da);
tacs=[tac 0;0 tac];
sys = series(tacs,ss2,[1,2],[1,2]);
tsenvel = tf(50,[1 50]);
tsenz = tf(50,[1 50 ]);
tsenq = tf(50,[1 50]);
tsens = [tsenq 0 0;0 tsenz 0;0 0 tsenvel];% 0;0 0 0 tsenvel];
[ass,bss,css,dss] = ssdata(tsens);
sysnew = series(sys,tsens);
[Ap(:,:,i),Bp(:,:,i),Cp(:,:,i),Dp(:,:,i)] = ssdata(sysnew);
%--------Loop Shaping Weights-------------%
wn1=400*[0.009 1];
wd1=[1 0.0001];
wd2=[1 0.00001];
wn2=5*[1 0.6]
w1 = [tf(wn1,wd1) 0;0 tf(wn2, wd2)];
w21=1;w22=1;w23=1;
w2 = [w21 0 0;0 w22 0;0 0 w23];
%-------------------------------------------------------
loopsys=ss(Ap(:,:,i),Bp(:,:,i),Cp(:,:,i),Dp(:,:,i));
gaug(:,:,i)=series(w1,series(loopsys,w2));
[Apw(:,:,i),Bpw(:,:,i),Cpw(:,:,i),Dpw(:,:,i)]=ssdata(gaug(:,:,i));
gaugsys(:,:,i)=pck(Apw(:,:,i),Bpw(:,:,i),Cpw(:,:,i),Dpw(:,:,i));
 %---------calculate the coprime factorisation
    [A,B,C,D]=unpck(gaugsys(:,:,1));
    [rowd,cold]=size(D);
    Rt=eye(rowd)+D*D';
    R=eye(cold)+D'*D;

    [rowp,colp]=size(B);

    setlmis([])
     P=lmivar(1,[rowp 1]);
     Z=lmivar(1,[rowp 1]);

   for i=1:11
        [A,B,C,D]=unpck(gaugsys(:,:,i));
       jj=newlmi;
       lmiterm([jj,1,1,P], 1,A,'s');
       lmiterm([jj,1,1,0], -C'*C);
       lmiterm([jj,2,1,P],B',1);
       lmiterm([jj,2,1,0],-D'*C);
       lmiterm([jj,2,2,0], -R);
   end
jj=newlmi;
lmiterm([-jj 1 1 Z],1,1);
lmiterm([-jj 2 1 0],1);
lmiterm([-jj 2 2 P],1,1);
lmisys=getlmis;
c = mat2dec(lmisys,zeros(rowp), eye(rowp));
options = [1e-2 200 1e5 10 0]
[copt,xopt] = mincx(lmisys,c,options);

Popt=dec2mat(lmisys,xopt,1);
Zopt=dec2mat(lmisys,xopt,2);
LL=-(B*D'+inv(Popt)*C')*inv(Rt);
[rowc,colc]=size(Cpw(:,:,1));
%---contruct the copirme factorisation design structure
for i=1:11
```

```
Acop=Apw(:,:,i);
b1=-LL*sqrt(Rt); b2=Bpw(:,:,i);
Bcop=[-LL*sqrt(Rt) Bpw(:,:,i)];
c1=[zeros(2,colc);Cpw(:,:,i)];
c2=Cpw(:,:,1);
Ccop=[c1;c2];
d11=[zeros(2,3);sqrt(Rt)];
d12=[eye(2);Dpw(:,:,i)];
d21=sqrt(Rt);
d22=[Dpw(:,:,i)];
Dcop=[d11 d12; d21 d22]
Gcop(:,:,i)=pck(Acop,Bcop,Ccop,Dcop);
end
vlpv=[];
for i=1:11
    vlpv=[vlpv ; Gcop(:,:,i)]
end
iv=1:11
vlpv=vpck(vlpv,iv);
Vel=Vel'
viv=vpck(Vel,iv);
```

controller synthesis

```
vparm=viv;
tstart = cputime;
nmeas=6;% number of measure
nctrl=2; % number of control
%calcuate the basis functin, parameter dependent lyapunov function
fdef=[1]; gdef=[1;  parm; parm^2]
gfdef=[0];ggdef= [ 0; 1; 2*parm];
```

```
%the parameter varying rate
vnu=[-10 10];
%Pole placement constraints
maxe=100; gest=20;
% retrieving information for system matrix and basis function.
[mtyp,sysrow,syscol,npts1]=minfo(vlpv);
viv=getiv(vlpv);
[typ,nparm,nbnds,npts6]=minfo(vnu);
npts6=npts1;
vnu=vpck(kron(ones(npts6,1),vnu),viv);
%checking the consistency of the basis function.
[typ,nbasisx,xcol,npts2]= minfo(fparm);
[typ,nbasisy,ycol,npts3]= minfo(gparm);
grid_pts=npts1;%
%Identify the parameters rate bound
igradf = any(vunpck(gradf)); % checking the parameter with 0 entries
igradg = any(vunpck(gradg)); % checking the parameter with 0 entries
ivun = any(vunpck(vtp(vnu)));% checking parameter without rate bound as 0
parmx = igradf & ivun
parmy = igradg & ivun
nparmx = sum(parmx);
nparmy = sum(parmy);
if nparmx>0
    nvertx=nbnds;
else
    nvertx=2^nparmx;
end
if nparmy>0
    nverty=nbnds;
else
    nverty=2^nparmy
```

```
end
% Get matrix containing all combination of vertices
combmatx=corners(nparmx);
combmaty=corners(nparmy);
sys=xtracti(vlpv,1,1);
[systype,no,ni,nx]=minfo(sys);
ne = no-nmeas;
ne1 = ne-nctrl;
nd = ni-nctrl;
nd1 = nd-nmeas;
ndecvar = nx*(nx+1)* (nbasisx+nbasisy)/2 +1
nvarxy = nbasisx + nbasisy;
% Setup lmi matrix variable
setlmis([])
for i=1:nbasisx
    X=lmivar(1,[nx,1]);
end
for i=1:nbasisy
    Y=lmivar(1,[nx 1]);
end
gamma=lmivar(1,[1 0]);
%----LOOP for GRID POINT Adding data to LMIS
for i=1:grid_pts
    % getting data for basis function, gradient and rate bound
    fdat=xtracti(fparm,i,1);
    gdat=xtracti(gparm,i,1);
    gfdat=xtracti(gradf,i,1);
    ggdat=xtracti(gradg,i,1);
    nu=xtracti(vnu,i,1);
    % Get state-space system matrices for grid point i
    sys = xtracti(vlpv,i,1);
    [a, b1, b2 c1, c2, d11]= transf(sys, nmeas, nctrl);
    [trow,tcol] = size(d11);
    if min(eig(eye(tcol)-d11'*d11)) <= 0
        disp('I - D11*D11 < 0');
    end
    b11 = b1(:,1:nd1);
    b12 = b1(:,nd1+1:nd);
    c11 = c1(1:ne1,:);
    c12 = c1(ne1+1:ne,:);
    d1111 = d11(1:ne1,1:nd1);
    d1112 = d11(1:ne1,nd1+1:nd);
    d1121 = d11(ne1+1:ne,1:nd1);
    d1122 = d11(ne1+1:ne,nd1+1:nd);
    ahat = a-b2*c12;
    atld = a-b12*c2;
    b1hat = b1-b2*[d1121 d1122];
    c1tld = c1-[d1112;d1122]*c2;
    indx = (nvertx+nverty+1)*(i-1);
    indx1 = indx+nverty;
    indx2 = (nvertx+nverty+1)*i;
    %-----------------setting up LMIs for Y
   if ne1 > 0
        ezmaty = [eye(nx) zeros(nx,ne1)];
    else
        ezmaty = 1;
    end
    for j = 1:nverty
        for k = 1:nbasisy
            lmiterm([indx+j 1 1 k+nbasisx],gdat(k)*[ahat;c11],ezmaty,'s');
            ly = 0;
            for l = 1:nparm
            if parmy(l) ~= 0
                ly = ly + 1;
                parmvl = (nu(l,1)+nu(l,2))/2 + combmaty(j,ly)...
                    *(nu(l,1)-nu(l,2))/2;
                if abs(ggdat(k,l)) > eps & abs(parmvl) > eps
                    lmiterm([indx+j 1 1 k+nbasisx],-parmvl*ggdat(k,l).
                        *ezmaty',ezmaty);
                end
            end
        end
    end
    lmiterm([indx+j 1 1 nvarxy+1],-1,daug(b2*b2',eye(ne1)));
     lmiterm([indx+j 1 1 nvarxy+1],-(Vt(i)),daug(b2*b2',eye(ne1)));
    lmiterm([indx+j 2 1 0],[b1hat' [d1111 d1112]']);
    lmiterm([indx+j 2 2 nvarxy+1],-1,1);
    lmiterm([indx+j 2 2 nvarxy+1],-(Vt(i)),1);
end
%------- setting up LMIs for X
if nd1 > 0
    ezmatx = [eye(nx) zeros(nx,nd1)];
else
    ezmatx = 1;
end
for j = 1:nvertx
    for k = 1:nbasisx
       lmiterm([indx1+j 1 1 k],fdat(k)*[atld';b11'],ezmatx,'s');
       lx = 0;
        for l = 1:nparm
            if parmx(l) ~= 0
                lx = lx + 1;
                    parmvl = (nu(l,1)+nu(l,2))/2 + combmatx(j,lx)*...
```

```
                            (nu(l,1)-nu(l,2))/2;
                    if abs(gfdat(k,l)) > eps & abs(parmvl) > eps
                        lmiterm([indx1+j 1 1 k],parmvl*gfdat(k,l)*...
                            ezmatx',ezmatx);
                    end
                end
            end
        end
        lmiterm([indx1+j 1 1 nvarxy+1],-1,daug(c2'*c2,eye(nd1)));
        lmiterm([indx1+j 1 1 nvarxy+2],-(Vt(i)),daug(c2'*c2,eye(nd1)));
        lmiterm([indx1+j 2 1 0],[c1tld [d1111;d1121]]);
        lmiterm([indx1+j 2 2 nvarxy+1],-1,1);
        lmiterm([indx1+j 2 2 nvarxy+2],-(Vt(i)),1);
    end
    %-------setting up LMIs  for both X and Y---
    delt=1e-6
    for k=1:nbasisy
        if abs(gdat(k)) >eps
            lmiterm([-indx2 1 1 k+nbasisx], gdat(k), 1);
        end
    end
    for k=1:nbasisx
        if abs(gdat(k)) > eps
            lmiterm([-indx2 2 2 k], fdat(k),1);
        end
    end
    lmiterm([-indx2 2 1 0],1);
    lmiterm([indx2 1 1 0], delt);
    lmiterm([indx2 2 2 0],delt);
end
%-----extra LMIS for pole placement contraints--------
```

```
   for k=1:nbasisy
    if abs(gdat(k)) > eps
     lmiterm([-indx2 1 1 k+nbasisx],gdat(k)/(2*maxe),ahat','s');
     lmiterm([-indx2 2 1 k+nbasisx],-gdat(k)/(2*maxe*gest)*c11'*c11,1);
    end
   end
   for k=1:nbasisx
    if abs(fdat(k)) > eps
     lmiterm([-indx2 2 2 k],fdat(k)/(2*maxe),atld,'s');
     lmiterm([-indx2 2 1 k],-fdat(k)/(2*maxe*gest),b11*b11');
    end
   end
   lmiterm([-indx2 2 2 nvarxy+1],-1/maxe,c2'*c2);
   lmiterm([-indx2 1 1 nvarxy+1],-1/maxe,b2*b2');
   lmiterm([-indx2 2 1 0],(b12*c2+b2*c12)'/(2*maxe));
  end
%%%%%%%%%%%%%%%%end of grid point loop%%%%%%%%%%%%%%%%%%%%55
lmis=getlmis;
cobj0=zeros(nx*(nx+1)/2*nvarxy,1)
cobj=[cobj0;1];
decinit=[]
%Call MINCX to solve the convex optimization
[copt,xyopt]=mincx(lmis,cobj,[1e-2 200 1e7 10 0], decinit ,0);
if isempty(copt)
    disp(['sorry, NO feasible solution found.']);
end
gam1 = xyopt(size(xyopt,1)-1);
gam2= xyopt(size(xyopt,1));
xmat=[];
ymat=[];
for i=1:nbasisx
```

```
    xmat=[xmat; dec2mat(lmis, xyopt,i)]
end
for i=1:nbasisy
    ymat=[ymat; dec2mat(lmis, xyopt, i+nbasisx)];
end
xmat=vpck(xmat,[1:nbasisx]')
ymat=vpck(ymat,[1:nbasisy]');
gam=[]
for i=1:11
    gam=[gam;gam1+Vt(i)*gam2]
end
gam=vpck(gam,Vt');
```

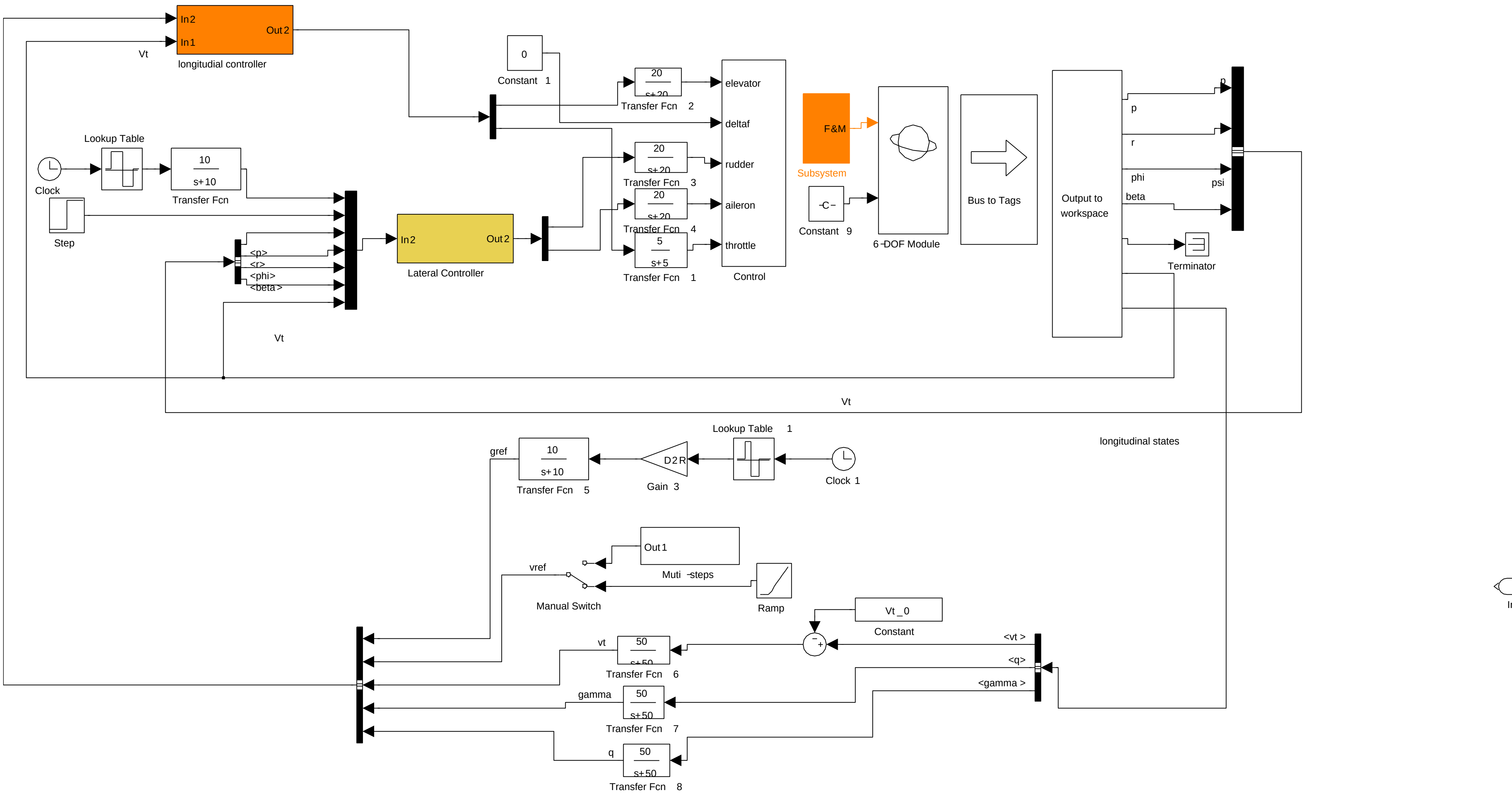
Controller construction

```

[typ,sysrow,syscol,npts1]= minfo(vlpv);
[typ, row1, col1,npts1]= minfo(vlpv);
viv=getiv(vlpv);
[typ,row0, col0,npts0]= minfo(gam);
gam=vpck(gam*ones(npts1,1),viv);
npts0=npts1;
[typ,xrow,xcol, numx]= minfo(xmat);
[typ, yrow, ycol, numy]= minfo (ymat);

grid_pts=npts1;
sys=xtracti(vlpv, 1,1);
[systype,no,ni,nx]=minfo(sys);
nx1= xrow;
nx2=nx-nx1;
ny1=nmeas-nx2;
```

```
ny2=nx2;
ne=no-nmeas;
ne1=ne-nctrl;
nd=ni-nctrl;
nd1=nd-nmeas;
sysrows=nx1+nctrl+1   % rows of the controller
syscols=nx1+nmeas+1   % cols of the controller ...
clprows=(nx+nx1)+ne+1 % rows of the clp system...
clpcols=(nx+nx1)+nd+1 % cols of the clp systems...
vctrl= zeros(sysrows*grid_pts,syscols);
vclp = zeros(clprows*grid_pts,clpcols);   % the intial zero matrices for pa

%Form the lyapunov matices for parameter i
for i=1:grid_pts
    X=zeros(nx1,nx);
    Y=zeros(nx);
    gamma= xtracti(gam,i,1);
    fvec=xtracti(fparm,i,1);
    gvec=xtracti(gparm,i,1);
    for k=1:nbasisx
        X= X + fvec(k) * xtracti(xmat,k,1);
    end

    for k=1:nbasisy
        Y = Y + gvec(k) * xtracti(ymat,k,1);
    end
    Y11=Y(1:nx1,1:nx1);
    Y12=Y(1:nx1,nx1+1:nx);
    Y22=Y(nx1+1:nx,nx1+1:nx);
    Yi=inv(Y);
    Yi11=Yi(1:nx1,1:nx1);
```

```
Yi12=Yi(1:nx1,nx1+1:nx);
Yi22=Yi(nx+1:nx,nx+1:nx);
X11=X(:,1:nx1);
X12=X(:,nx1+1:nx);
Z11=X11-Yi11;
Z12=X12-Yi12;
X22=Z12'/Z11*Z12+Yi22;
X=[X11 X12; X12' X22];
N=[eye(nx1);Z12'/Z11];
M=-Y*[Z11; Z12'];
N1=N(1:nx1, :);
N2=N(nx1+1:nx,:);
M1=M(1:nx1,:);
M2=M(nx1+1:nx,:);
%Get state-space data from parm_i
sys=xtracti(vlpv,i,1);
[A,Bd,Bu,Ce,Cy,Ded,Deu,Dyd,Dyu,r12inv,r21inv,q12,q21,flag]= ...
    transfr(sys,nmeas,nctrl,nx2);
   Cy1=Cy(1:ny1,:);
   Dy1d=Dyd(1:ny1,:);
        dk1 = zeros(nctrl,ny1);
        F = -(gamma*Bu'/Y + Deu'*Ce);
        L = -(X\Cy'*gamma + Bd*Dyd');
    akhat = -X*(Bd+L*Dyd)*Bd'/gamma - Ce'*(Ce+Deu*F)*Y/gamma;
    aktld = (A'+X*(A+Bu*F+L*Cy)*Y-akhat);
    akt(:,:,i)=aktld;
    dcbak = [F*Y; -N1\aktld]/([Y12 M1; Y22 M2]');
    ak = dcbak(nctrl+(1:nx1),ny2+(1:nx1));
    bk = N1\X(1:nx1,:)*L + [zeros(nx1,ny1) dcbak(nctrl+(1:nx1),1:ny2)]
    ck = dcbak(1:nctrl,ny2+(1:nx1));
    dk = [dk1, dcbak(1:nctrl,1:ny2)];
   %----inversion of the norm preserving transformation----------------
    dk = r12inv*dk*r21inv';
    ck = r12inv*ck;
    bk = bk*r21inv';
   %-------------------------------------------------------------------
    if any(any(Dyu))
        ck = (eye(nctrl) + dk*Dyu)\ck;
        dk = (eye(nctrl) + dk*Dyu)\dk;
        ak = ak - bk*Dyu*ck;
        bk = bk - bk*Dyu*dk;
    end

    ctrl = pck(ak,bk,ck,dk);
    clp = starp(sys,ctrl);
    vctrl((i-1)*sysrows+1:i*sysrows,:) = ctrl;
    vclp((i-1)*clprows+1:i*clprows,:) = clp;
end
    vclp=vpck(vclp, viv);
    vctrl=vpck(vctrl, viv);

 for i=1:11
[ak(:,:,i),bk(:,:,i),ck(:,:,i),dk(:,:,i)]=unpck(xtracti(vctrl,i,1))
end
```

Simulink diagram



In2
In1
Out2
Vt
longitudial controller
0
Constant 1
20
s+20
Transfer Fcn 2
elevator
deltaf
rudder
aileron
throttle
Control
Lookup Table
10
s+10
Transfer Fcn
Clock
Step
<p>
<r>
<phi>
<beta>
Vt
In2
Out2
Lateral Controller
20
s+20
Transfer Fcn 3
20
s+20
Transfer Fcn 4
5
s+5
Transfer Fcn 1
F&M
Subsystem
-C-
Constant 9
6-DOF Module
Bus to Tags
Output to workspace
p
p
r
phi
psi
beta
Terminator
Vt
Lookup Table 1
longitudinal states
gref
10
s+10
Transfer Fcn 5
D2R
Gain 3
Clock 1
Out1
Muti -steps
vref
Manual Switch
Ramp
Vt_0
Constant
vt
50
s+50
Transfer Fcn 6
<vt>
<q>
<gamma>
gamma
50
s+50
Transfer Fcn 7
q
50
s+50
Transfer Fcn 8
I

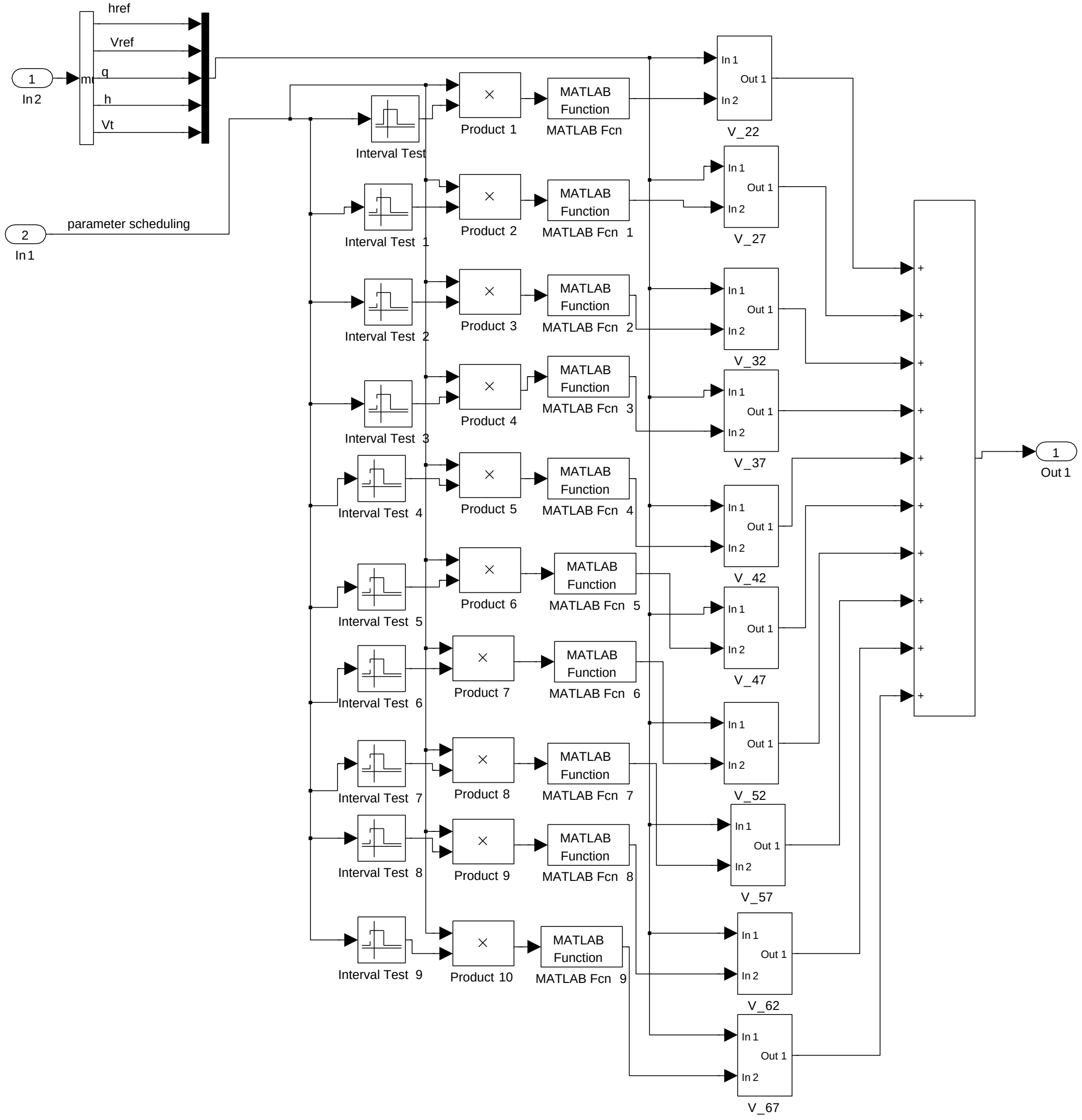


Figure 7.2: Sub-interconnection of longitudinal/lateral controller

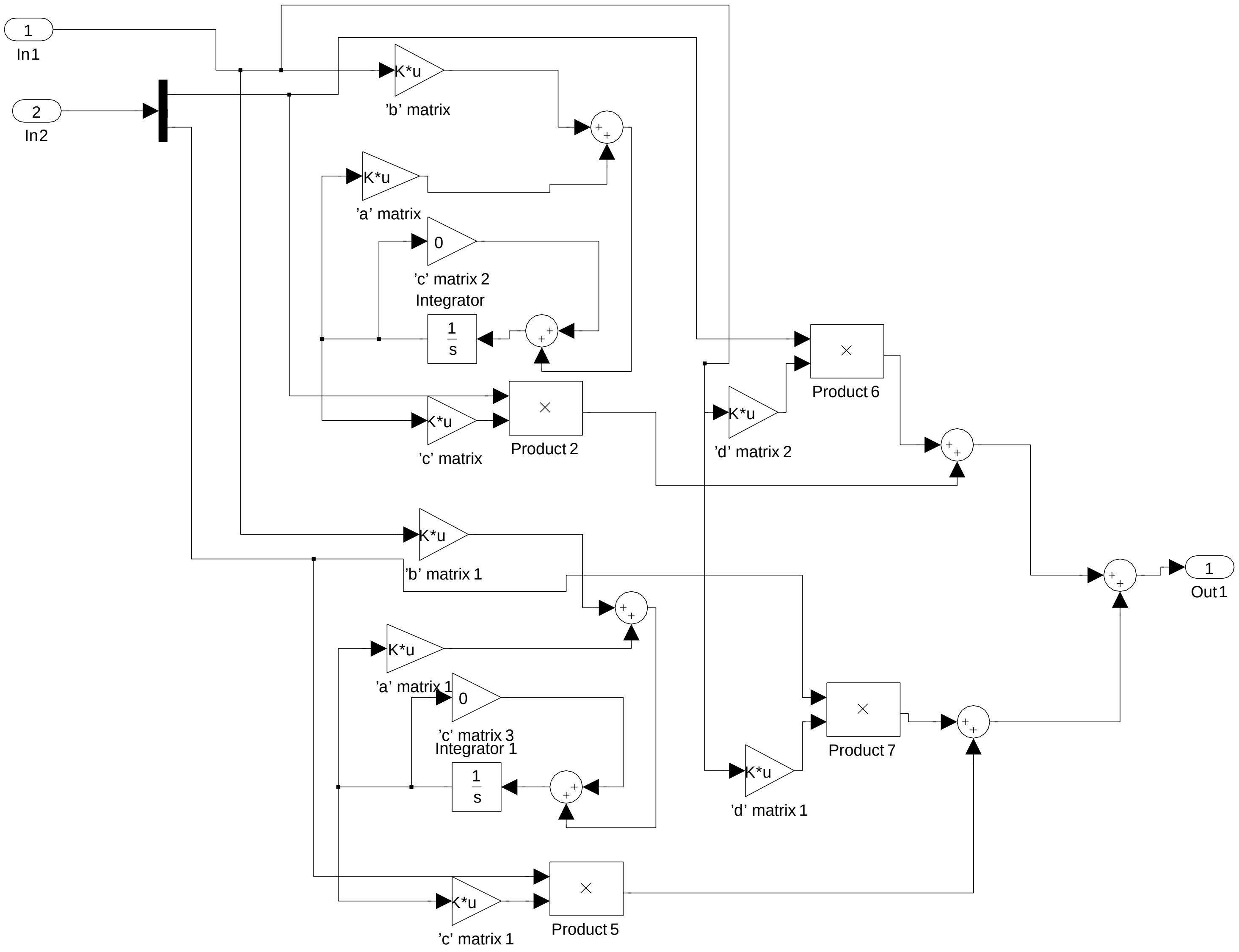


Figure 7.3: Sub-interconnection of V_22–V_67

Bibliography

[1] N. Sivashankar and I. Kaminer, “Design, Analysis and Hardware-in-the-loop Simulation of a MIMO Controller for VTOL Unmanned Aerial Vehicle using H_∞ Synthesis,” *Proceedings of the American Control Conference*, Baltimore, 1994.

[2] J. Wang, V. Patel and A. Woolsey, “L_1 Adaptive Control of a UAV for Aerobiological Sampling,” *Proceedings of American Control Conference*, 2007

[3] S. Boyd, L. EL Ghaoui, E. Feron and V. Balakrishnan ,“Linear Matrix Inequalities in System and Control Theory, ” Siam, Philadephia, 1994.

[4] A.S.Householder, “The Theory of Matrices in Numerical Analysis,” Dover, New York, 1974.

[5] P. Gahinet and P. Apkarian, “A Linear Matrix Inequality Approach to H_∞ Control,” *International Journal of Robust and Nonlinear Control*, Vol. 4, pp. 883–887, 1994.

[6] Y. Nesterov and A. Nemirovsky, “Interior Point Polynomial Method in Convex Programming,” *Studies in Applies Mathematrics*, Vol. 13, pp. 22–49, 1994.

[7] A. Packard, “Gain Scheduling via Linear Fractional Transformations,” *System & Control Letters*, Vol. 22, pp. 79-92, 1994.

[8] G. Scorletti and L. El Ghaoui, “Improved Linear Matrix Inequality Conditions for Gain Scheduling,” *Proceedings of the 34th IEEE Conference on Decision and Control*, Vol. 4, pp. 3626–3631, 1995.

[9] S. Shahruz and S. Behtash, “Design of Controllers for Linear Parameter Sysntems by Gain Scheduling Technique,” *Journal of Mathematrical Analysis and Applications*, Vol. 168, pp. 195–217, 1992.

[10] K. Glover and D. McFarlane,“Robust Stabilization of Normalized Copriem Factor Plant Descriptions with H_∞ Bounded Uncertainty,” *IEEE Transactions on Automatic Control*,” Vol. 34, pp. 821–830, 1989.

[11] D. Gu, P. H. Petkov and M. Konstantinov, “Robust Control Design with Matlab,” Springer, 2002.

[12] D. Bates and I. Postlethwaite, Robust Multivariable Control of Aerospace Systems,” Delft Unverisity Press, 2002.

[13] R.A.Hyde, “H_∞ Aerospace Control Design: a VSTOL Flight Application,” Springer, 1996.

[14] F.M. Callier and C. Desoer, “Linear System Theory”, Springer-Verlag, New York, 1991.

[15] F. Wu, “Control of Linear Parameter Varying Systems”, *Ph.D dissertation*, Department of Mechanical Engineering, University of California at Berkeley, 1995.

[16] P. Apkarian and P. Gahhinet , “A Convex Characterization of Gain-scheduled H_∞ Controllers”, *IEEE Trans.on Automatic Control*, Vol. 40, pp. 853–864, 1995.

[17] P. Apkarian ,P. Gahinet and G. Becker, “Self-scheduling H_∞ Control of Linear Parameter-varying Systems: A Design Example”, *Automatica*, Vol. 31(1), pp. 1251–1261, 1995.

[18] P. Apkarian and J. Adams, “Advanced Gain-Scheduling Techniques for Uncertain System”, *IEEE Transation on Automatic Control*, Vol. 6(1), pp. 21–32, 1998.

[19] P. Gahinet and P. Apkarian, “Affine Parameter Depdnent Lyapunov Functions and Real Parameteric Uncertainty”, *IEEE Transaction on Automatic Control*, Vol. 41, pp. 436–442, 1998.

[20] S. Bennani, D.M.C. Willemsen and C. W. Scherer, “Robust Control of Linear Parametrically Varying Systems with Bounded Rates”, *Journal of Guidance, Control and Dynamics*, Vol. 21(6), pp. 916–922, 1998.

[21] C. van Etten, G. J. Balas and S. Bennani, “Linear Parametrically Varying Integrated Flight and Structural Mode Control for a Flexible Aircraft”, *AIAA Guidance, Navigation and Control Conference and Exhibit*, Vol. 2(A99–36576 09–63), Portland, 1999.

[22] M. J. Siersma, R. van der Weerd and S. Bennani, “Robustness Analysis of a Gain-scheduled Flight Control System using Integral Quadratic Constraints”, *AIAA Guidance, Navigation and Control Conference and Exhibit*, Denver, 2000.

[23] F. Wu, X. Yang, A. Packard and G. Becker, “Induced L_2 norm Control for LPV system with Bounded Parameter Variations Rates”, *Proceedings of American Control Conference*, Seattle, WA, pp. 436–4422, 1995.

[24] M. Chilali and P. Gahinet , “H_∞ Design with Pole Placement Constraints: An LMI Approach ”, *IEEE Transaction on Automatic Control*, Vol. 41(3), pp. 358–366, 1996.

[25] M. Chilali, P. Gahinet and P. Apkarain, “Robust Pole Placement in LMI Region”, *IEEE Transaction on Automatic Control*, Vol. 44(12), pp. 2257–2269, 1999.

[26] M. Safonov ,D. Limbeer and R. Chiang , “Simplifying the H_∞ Theory via Loop Shifting. Matrix Pencil, and Descriptor Concepts,” *International Journal of Control*, Vol. 50, pp. 2467–2488, 1989.

[27] S.T. Lin and C. Lee, “Multivairiable Control of the Turbojet Engine for Full Flight Envelope Operation,” *Jouranl of Guidance, Control and Dynamics*, Vol. 19, pp. 913–920, 1996.

[28] R. A. Nichols, R. T. Reichet and W. J. Rugh, “Gain Scheduling for H_∞ Controllers: a Flight Control Example”, *IEEE Transactions on Control System Technology*, vol. 1, pp. 66–79, 1993.

[29] R.A. Hyde and K. Glover, "The Application of Scheduled H_∞ Controllers to a VSTOL Aircraft", *IEEE transactions on Automatic Control*, Vol. 38, pp. 1021–1039, 1993.

[30] J. S. Shamma and M. Athans,"Gain Scheduling: Potential Hazards and Possible Remedies", *IEEE Control Systems Magazine*, Vol. 12 ,1992.

[31] G. Becker and A. Packard, "Robust Performance of Linear Parametrically Varying System using Parameter-Dependent Linear Feedback," *System & Control Letters*, vol. 23, pp. 205–215, 1994.

[32] G. Becker, "Quadratic Stability and Performance of Linear Parameter Depedent Systems," *PhD Thesis*, University of California, Berkeley, 1993.

[33] G. Becker, "Additional Results on Parameter-dependent Controllers for LPV systems," *IFAC World Congress*, San Fransisco, CA, 1996.

[34] L. Lee and M. Spillman, "A Parameter-Dependent Performance Criterion for Linear Parameter-Varying systems", *Proceedings of the IEEE Conference on Decision and Control*, 1997.

[35] L. Lee, "Identification and Robust Control of Linear Parameter Varying Systems," PhD Thesis, University of California, Berkeley, 1997.

[36] L. Lee and M. Spillman, "Robust, Reduced-Order, Linear Parameter-Varying Flight Control for An F-16", *AIAA Guidance, Navigation and Control Conference*, 1997.

[37] D. McFarlane and K. Glover, "A Loop Shaping Design Procedure using H_∞ Syntheisis", *IEEE Trans.on Automatic Control*, Vol. 37(6), pp. 759–769, 1992.

[38] J.F. Whidborne, I. Postlethwaite and Da-wei Gu, "Robust Controller Design using H_∞ Loop Shaping and the Method of Inequlaities, *IEEE Transaction on Control Systems Technology*, Vol. 2(4), pp. 455–461, 1994.

[39] G. Papageorgiou and K. Glover, “Robust Control System Design H_∞ Loop Shaping and Aerospace Application”, *PhD Thesis*, 1997.

[40] S. Patra , S. Sen and G. Ray, “Design of Static H_∞ Controller in Four-block Framework using LMI Approach”, *Automatica*, pp. 2214–2220 2008.

[41] M. Jung and K. Glover, “Calibratable Linear Parameter-Varying Control of a Turbocharged Diesel Engine”, *IEEE Transaction on Automatic Control*, Vol. 14(1), pp. 44–56, 2006.

[42] E. Prempain , “Gain-scheduling H_∞ Loop Shaping Control of Linear Parameter Varying Systems”, *ROCOND06*, Toulouse, France, 2006.

[43] E. Prempain , “Coprime Facotrizations for Parameter Dependent Systems”, *ROCOND06*, Toulouse, France, 2006.

[44] E. Prempian,I. Postlethwaite, “Static H_∞ Loop Shaping Control of a Fly-by-wire Helicopter”, *Automatica*, Vol. 41, pp. 1517–1528, 2005.

[45] S. Skogestad, I. Postlethwaite, “Multivariable Feedback Control: Analysis and Design”, Wiley, 1996.

[46] P. Gahinet, P. Apkarian and M. Chilali, “Parameter-dependent Lyapunov Functions for Real Parametric Uncertainty”, *IEEE Transaction Automatic Control*, Vol. 41, pp. 436–442, 1996.

[47] E. Feron, P. Apkarian and P, Gahinet., “Analysis and Synthesis of Robust Control Systems via Parameter-dependent Lyapunov Functions”, *IEEE Transation on Automatic Control*, Vol. 41, pp. 1041–1046, 1996.

[48] S. Ghersin and S. Ricardo, “LPV Control of 6DOF Vechicle” ”, *IEEE Trans.on control system technology*, Vol. 10(6), pp. 883–887, 2002.

[49] S. Gutman and E.I. Jury, ”A General Theory for Matrix Root Clustering in Subregions of the Complex Plant”,*IEEE Trans.on control system technology*, Vol. 26 , pp. 853–863, 1981.

[50] K. Natesan, I. Postlethwaite and Da-wei. Gu, "Design of Static H_∞ Loop Shaping Controllers for LPV Plant", *FLAVIIR Project Report*,2006.

[51] K. Zhou, J. Doyle and K. Glover, "Robust and Optimal Control", Englewood Cliffs, NJ: Prentice-Hall Internationl.

[52] P. Gahinet, "Explicit Controller Formulas for LMI-based H_∞ Synthesis", *Automatica*, Vol. 32(7), pp. 1007–1014, 1996.

[53] M. Tucker and D. Walker, "RCAM Design Challenge: An H_∞ Approach.", GARTEUR Report/TP-088-21.

[54] M. Safonov ,D. Limbeer and R. Chiang , "Simplifying the H_∞ Theory via Loop shifting. Matrix Pencil, and Descriptor concepts," *International Journal of Control*, vol. 50, pp. 2467–2488, 1989.

[55] A. Buonanno , "Flight Dynamics Model of the Flying demonstrator UAV," *Aerodynamics report FLAV-A01-003*, 2005.

[56] A. Buonanno, "An Aerodynamic Simulation Model of the Eclipse UAV," *Aerodynamics report FLAV-A01-002*, 2005.

[57] M.V Cook, "Flight Dynamics Principles," Edward Arnold, September, 1997.

[58] B. Steven and F. Lewis, "Aircraft Control and Simulation", *Wiley*, 1992.

[59] G. Balas, A. Packard and J. Renfrow, "On the Design of the LPV controller for the F-14 Aircraft Lateral-directional axis during powered approach," *American control conference*, pp. 123–127, 1997.

[60] A. Marcos and G. Balas, "Development of Linear Parameter Varying Models for aircraft",*Journal of Guidance,Control and Dynamics*, Vol. 27(2), pp. 218–228, 2004.

[61] K. Natesan, D. Gu, I. Postlethwaite and J. Chen, "Design of Flight Controllers based on Simplified LPV model of a UAV", *IEEE Conf. Decision and Control*, 2006.

[62] B. Liu and F. Wu, “Probabilistic Robust Linear Parameter Varying Control of an F-16 Aircraft”, *Journal of Guidance*, Vol. 29(6), pp. 1454–1460, 2006.

[63] P. Gahinet, A. Nemirovski and M. Chilali, “LMI Control Toolbox”, The MathWorks.